UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA

CÁTEDRA EMT/UNED

DOCTORAL THESIS

Spatio-temporal neural models for sustainable mobility and air quality

Author:
Rodrigo de Medrano

Supervisor:
José Luis Aznarte, PhD

A thesis submitted in fulfillment of the requirements for the degree of PhD. in Intelligent Systems

October 5, 2021

“The happiness of your life depends upon the quality of your thoughts.”

Marcus Aurelius

UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA

Abstract

Computer Science faculty
Department of Artificial Intelligence

PhD. in Intelligent Systems

Spatio-temporal neural models for sustainable mobility and air quality

by Rodrigo de Medrano

[*ENG*]

Although temporal and spatial series have been broadly studied on their own, spatio-temporal systems remain still an open research field. Given that a vast number of natural and human processes fit into a spatio-temporal domain, new tools and methodologies are rising. Among them, neural networks have shown their ability to perform very well in high dimensional and nonlinear environments, which makes this kind of models very attractive for spatio-temporal series forecasting.

Within this context, sustainable mobility and air quality are growing in importance due to its high impact on public health, logistic, and economy. Due to these facts, today and more than ever, it is desirable to develop tools that let us optimize and predict processes along these topics. Thus, this thesis focuses its efforts in tackling these issues. Concretely, we advanced the general understanding of spatio-temporal convolutional networks while proposing a new model for interpretable traffic forecasting and a new approach to predict air quality (which is in use by Madrid authorities).

[*ESP*]

Aunque las series temporales y espaciales han sido ampliamente estudiadas por separado, los sistemas espacio-temporales propiamente dichos siguen siendo un campo abierto de investigación. Dado que un gran número de procesos naturales y humanos encajan en un dominio espacio-temporal, el desarrollo de nuevas herramientas y metodologías destinados a caracterizarlos es de vital importancia. Entre ellas, las redes neuronales tienen la capacidad de desenvolverse bien en entornos altamente dimensionales y no lineales, lo que hace que este tipo de modelos sean muy atractivos para la predicción de fenómenos espacio-temporales.

Por otro lado, es un hecho que hoy en día la movilidad sostenible y calidad del aire son campos que están adquiriendo una importancia creciente debido a su gran impacto en la salud pública, logística y economía. Dentro de esta casuística es de gran interés desarrollar herramientas que nos permitan optimizar y predecir procesos en tales campos. Debido a ello, esta tesis centra todos sus esfuerzos en abordar cuestiones enmarcadas en dichas áreas de estudio. En concreto, se propone un nuevo modelo para la predicción del tráfico centrado en la mejora de la interpretabilidad de sistemas neuronales, se profundiza en el uso de redes convolucionales para resolver problemas espacio-temporales y se propone un nuevo marco para la predicción de la contaminación en Madrid que se encuentra actualmente operativo como la herramienta oficial del ayuntamiento.

Acknowledgements

Son, sin ningún género de dudas, muchas las personas que han influido de forma directa o indirecta a que esta tesis salga adelante, así como a que el resultado final de ella sea el que aquí se presenta. Por tanto, son muchas las personas que merecerían mención en estas líneas, aunque por razones obvias me es imposible aludir a todas. Desde ya, mis más sinceras disculpas, pues toda contribución, por pequeña que sea, ha sido determinante.

En primer lugar, y como no podía ser de otra manera, es obligado pero sobre todo un enorme placer mencionar al Dr. José Luis Aznarte. De este enorme proyecto me llevo muchas cosas, pero pocas comparables con su dirección y amistad. Gracias por saber guiarme sin por ello cortarme las alas de la creatividad. Por haber estado siempre disponible y atento a mis necesidades. Por confiar en mi. En definitiva, por haberme regalado la oportunidad de aprender y formarme científica y personalmente estos años a tu lado.

Obligado también es referenciar a mi madre y a mi padre: no hay personas para las que (por razones obvias, y no tan obvias) decir que sin ellos esto no habría sido posible sea más cierto.

A mi hermano Javier, quien me ha mostrado que a veces lo que necesitamos no es que remen en nuestra dirección, si no que dejes de remar y reflexiones sobre si la dirección de viaje está siendo la correcta.

También es importante darle protagonismo al resto de mi familia: porque sus consejos y puntos de vista me han ayudado en este proceso de maduración, en algunas ocasiones de forma mucho más profunda de la que probablemente crean.

A mis amigos y amigas, que han sido un pilar fundamental sobre el que reposar la carga que produce una aventura como esta. Aún más importante, por enseñarme que no es cuestión de que no se puedan conseguir logros lidiando uno mismo con todo, es cuestión de que no es necesario. Sin vuestra amistad, este desierto no habría tenido oasis. Especial mención a Nuria y Guillermo, quienes han tenido una implicación particularmente importante en el desarrollo de este trabajo.

A todos aquellos que han tenido implicación directa desde un punto de vista de desarrollo tanto científico como técnico. Decir que el trabajo en equipo multiplica las bondades y posibilidades de lo que se puede lograr es probablemente de una evidencia aplastante, pero precisamente por eso es tan importante apuntarlo: este trabajo no podría haber llegado hasta aquí sin vuestra ayuda.

Y por supuesto, no puedo dejar de reconocer este como un trabajo conjunto en el que existe otra figura fundamental a la que considero como "segunda autora" de esta tesis: Irene. Gracias por haber sido mi motor durante este largo y complejo camino. Sin ese genuino y desinteresado apoyo, la persona que estaría aquí hoy sería muy diferente. Este será siempre también logro tuyo.

Contents

Chapter 1

Introduction

Through this first chapter, a general vision of the problem will be offered and its importance will be highlighted. Motivation for the studied topic and general objectives are also discussed. In the end, there is a brief overview of the thesis.

1.1 Context

The ability to predict the future behaviour of a process or phenomenon is an idea that has constantly and universally seduced mankind throughout its history. With the development and advancement of several science disciplines and mathematics, new tools that allow modeling future states of dynamic systems have recently been proposed. Thus, in the last decades, time and spatial series have been broadly studied, but spatio-temporal systems that combine information about these two dimensions remain still an open field. From climate science and transportation systems to finance and economics, there are plenty of fields in which time and space might constitute two entangled dimensions of data, with one affecting the other and thus both being relevant for prediction. Given that a vast number of natural and human processes fit into a spatio-temporal domain, new tools and methodologies are rising in order to better analyse, model, forecast and, finally, understand our world.

Recently, the improvement in computational capabilities, the development of algorithms, and the tendency to record and store all the information that is captured have allowed data science to emerge as a fruitful field which a vast number of disciplines can take advantage of. Thus, there has been an increasing trend to develop and improve methodologies for gathering and using vast amounts of spatio-temporal data over the last years. Tailored to extract usable knowledge from these big data repositories, it is an extraordinary opportunity for developing proposals that aim to facilitate a shared understanding of the multiple relationships between the physical and natural environments and society. By contributing in this direction, it is possible to enrich plenty of services in many ways.

In the last decade, while machine learning has been widely used for, precisely, gaining insight into the aforementioned topics, there is still room for improvement in our understanding of the models and their applications. Among these models, neural networks have the ability to behave well in high

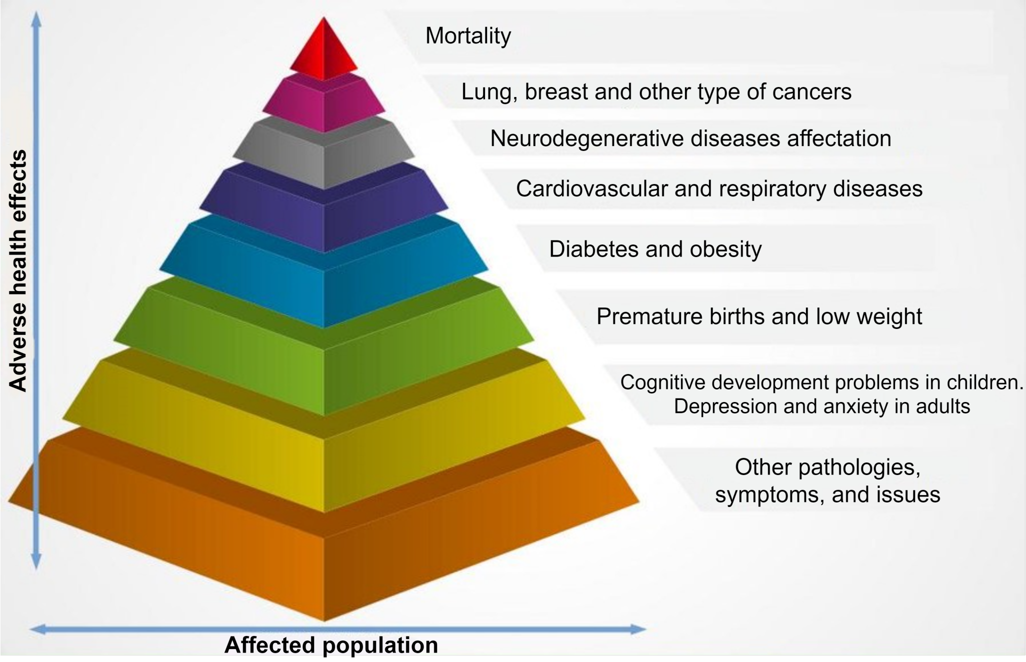

Mortality
Lung, breast and other type of cancers
Neurodegenerative diseases affectation
Cardiovascular and respiratory diseases
Diabetes and obesity
Premature births and low weight
Cognitive development problems in children. Depression and anxiety in adults
Other pathologies, symptoms, and issues
Adverse health effects
Affected population

FIGURE 1.1: Proportion of population affected and adverse effects of contamination in health [90].

dimensional and non-linear environments, which make this kind of models very attractive for solving problems and modelling systems belonging to a spatio-temporal domain.

Within this context, sustainable mobility and air quality are growing in importance due to their high impact on public health, logistic, and the economy. Thus, it is desirable to develop tools that let us optimize and predict spatio-temporal processes related with these fields. Concretely, this work focus on (although it is not restricted to) two of the main issues related to the aforementioned fields: traffic and air quality. As is commonly known, traffic is one of the main elements that produce pollution. In turn, pollution is considered as one of the main sources of unhealthiness and mortality nowadays (see figure 1.1), making it especially interesting to develop new methodologies and systems that help us to alleviate the negative effects they cause. Even outside the field of public health, these phenomena are highly interesting because of the impact they can have on both the individual and collective levels.

All things considered, it is not surprising that institutions from a diversity of fields are investing a large amount of resources to increase our understanding of these phenomena and to alleviate their consequences. It is precisely in this context that the Empresa Municipal de Transportes (public municipal mobility company) of Madrid, EMT, showing its commitment to combining mobility and sustainability, has funded the research presented herein.

1.2 Motivation

There are many reasons why developing an entire thesis on the use of neural networks for spatio-temporal series related to sustainable mobility and air quality forecasting is interesting.

On the one hand, it is crucial, and to some extent unavoidable, to talk about the importance that most of the elements that support this thesis have gained in recent years. Neural networks, which only 20 years ago were little more than a laboratory amusement, have become one of the main methodologies that govern not only Artificial Intelligence but most modern computational applications. There are plenty of examples where these models are growing in importance: images, videos, audio, natural language processing, signal processing, etc. A vast number of fields are being explored and exploited employing this paradigm, in many cases with increasing success, reaching milestones every few months.

Certainly, it is also well known that spatio-temporal processes are important *per se* since they model or govern a great variety of processes in which the human being is highly interested or whose dynamics deeply affect the way we live, plan, and relate to each other as discussed in the previous section.

Lastly, up to this point it is well known that air quality and sustainable mobility are issues with a social, economic, and sanitary repercussion of crucial importance, being one of the concerns with greater representation in the collective sensibility. Thus, and like with neural networks, a big amount of resources are being required and used to solve the problems associated with these fields.

We are therefore facing a perfect cocktail of elements and scientific momentum at the right time from an objective point of view, which could be a reason of sufficient significance on its own to motivate a thesis such as the one presented here.

However, it should be noted that the pillars of this work are not only of practical interest or importance. From a subjective point of view, there are several questions that this thesis addresses whose answers are of interest in their own right. Among them, one of particular importance and significance is that the way in which neural networks use spatio-temporal information is not well known yet. Also, traffic/mobility forecasting is recurrent when using spatio-temporal neural models, but new tendencies as graphs and attention mechanisms are becoming popular, gaining ground to become the pillars supporting modern intelligent transportation systems. At this point, it is not clear how far these new neural systems can go in this field. Lastly, pollution/air quality forecasting has been mainly tackled from a time series perspective, allowing us to lead the new trend of scientific research that seeks to further develop the spatial dimension in this particular field.

Thus, although the nature of this thesis is mainly engineering, the work delves also into more purely theoretical matters.

1.3 Objectives

Given what has been said, the objectives of this thesis could be summarised as follow:

- To understand, develop, and use neural models for spatio-temporal series forecasting.
- To apply these models in issues related to sustainable mobility and air quality in order to improve public health and mobility.

Specific objectives for several proposals that compose this thesis will be presented in their correspondent chapter.

1.4 Thesis overview and contributions

Given the broad and interdisciplinary perspective of this thesis, this work has been conceived as a compendium of projects with functional independence but with a common unifying thread. By doing so, we have had the opportunity to develop a varied and wide-ranging work that allows us to approach the problem from several points of view. Thus, this thesis starts from a pure spatio-temporal systems modeling approach, to then delve into the intricacies of how certain neural systems which are especially interesting for our field behave, and finally apply all the experience and material previously developed in a real application that is currently used officially by the municipality of Madrid. Regardless of the general tone of each project, we have always honoured, as scrupulously as possible, a methodology based on rigorous experimentation within data science.

Hence, this document is structured as follows. Chapter 2 contains an overview of neural methods that are commonly used throughout this thesis in particular, and in the field of spatio-temporal regression in general. Chapter 3 contains the first main project that sustains this document: the proposal of a new traffic prediction model based on attention mechanisms that takes special care of the ability to extract information related to the temporal part of the series through interpretability systems inherent to the aforementioned mechanisms.

After this, Chapter 4 focuses on the second main project. Specifically, it delves from a theoretical point of view into how convolution-based neural systems handle spatial information and refutes some classical ideas in their implementation when working with real spatio-temporal problems, pointing towards possible new proposals to be used in the future.

Chapter 5 presents a collaborative effort to build an operational air quality prediction system for the city of Madrid. This system combines the Bayesian statistical modelling developed by the team at Inverence with a deep neural network which, combined, allows the tool to be adapted to the pollution protocol of the city in such a way that it is functional and operational for forecasting and decision making in this important field of public health.

Chapter 6 and Appendix A work as related side projects but are slightly different from the general line of this thesis. In particular, a new system for estimating car accidents is proposed and an exhaustive analysis is made of the state of the art on the use of neural networks for the prediction and optimization of processes related to the COVID pandemic, respectively.

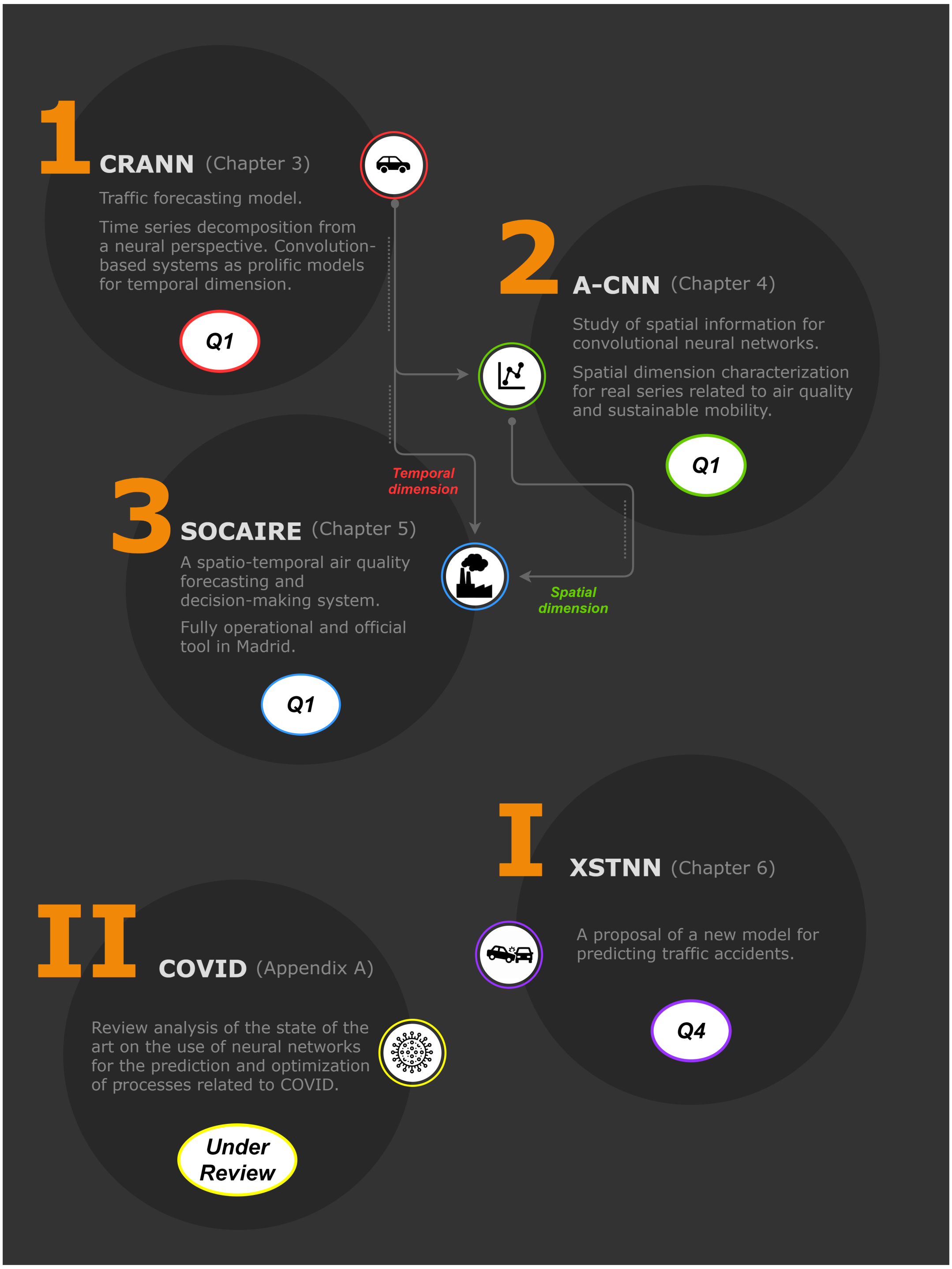
As the last part of this document, Chapter 7 is dedicated to summarize the main contributions of this work and to gather some future research lines that could be explored.

As can be inferred from the structure presented above, there is a clear path whose beginnings are of a purely theoretical and laboratory nature, but which has allowed the development of practical and complex systems of great potential for the health and quality of life of the population. Specifically, Chapter 3 and 4 projects develop, among other things, the understanding and knowledge of the use of neural networks for time and spatial series, always approaching the problem from a controlled point of view but taking care that it is as applicable as possible to real problems. Thus, the step taken in Chapter 5 is natural and shows how all the knowledge accumulated during the previous projects can be used when facing real life.

In this way, this thesis has provided the opportunity to go through a complete cycle of the scientific path. This path, which sometimes forgets that goes beyond the laboratory, does indeed start from the corroboration of hypotheses and the proposition of solutions. But it is when these solutions are applied in order to improve some aspect that affects society that a cycle is completed. Figure 1.2 summarises these ideas and puts them in context with this thesis in particular.

As stated previously, it is worth noting that this work has been developed under the financing and collaboration of the *Empresa Municipal de Transportes de Madrid* (EMT), without which this work would not have been possible. Also, during the process of elaboration of this document there has been the opportunity to co-direct two Master's theses, whose experience, besides being gratifying from a personal and laboral point of view, have resulted in direct material and inspiration for this thesis.

Finally, it is important to mention that scientific knowledge has been built around a temple that has falsifiability, reproducibility, and openness as its pillars. Therefore, whenever possible, an extra effort has been made to facilitate access to the computational tools developed as well as to the experiments carried out throughout this thesis, which can be found in the following link: https://github.com/rdemedrano. In all cases, this fact is noted in each article with its respective code where appropriate.



1
CRANN (Chapter 3)
Traffic forecasting model.
Time series decomposition from a neural perspective. Convolution-based systems as prolific models for temporal dimension.
Q1
2
A-CNN (Chapter 4)
Study of spatial information for convolutional neural networks.
Spatial dimension characterization for real series related to air quality and sustainable mobility.
Q1
Temporal dimension
3
SOCAIRE (Chapter 5)
A spatio-temporal air quality forecasting and decision-making system.
Fully operational and official tool in Madrid.
Q1
Spatial dimension
I
XSTNN (Chapter 6)
A proposal of a new model for predicting traffic accidents.
Q4
II
COVID (Appendix A)
Review analysis of the state of the art on the use of neural networks for the prediction and optimization of processes related to COVID.
Under Review

FIGURE 1.2: Research projects and their relation when correspond. Acronyms characteristic of each project are used and will be introduced in their respective chapters. In the figure it can be seen which acronym corresponds to which chapter/project. Ellipses show quartile of the journal for each project if published.

Chapter 2

Neural models for spatio-temporal series regression

This second chapter intends to be an introductory view of neural methods that are commonly used for spatio-temporal regression. Although the independent articles that form this thesis describe the methodology necessary for their understanding, it is summarised here through a brief explanation of those technicalities shared throughout the different papers that might be helpful for not deep learning practitioners. Sections 2.1 to 2.4 present a series of usual models, while Section 2.5 condenses general notes on the training and parameterization of the models seen before.

2.1 Feedforward neural networks

A feedforward neural network is a biologically inspired system that consists of a number of simple neuron-like processing units, organized in layers. They can model nonlinear processes based on the information collected by the input layer (which corresponds to the first one), propagating this information layer by layer establishing the relations between the inputs and the final layer called output layer. The larger the number of layers is, the deeper the network becomes, letting us model more complex relations and phenomena (and this fact would actually name the field as deep learning). Every unit in a layer is connected with all the units in the previous layer. These connections are not all equal: each connection will be represented by a different number which will be called weight. The weights encode the knowledge and transformations that perform a specific network. Each layer output (a) depends on the previous one (x) through a linear relation via the learned weight (W), and modulated by a non-linear function called activation function (g). Usually, a bias term (b) is added to the linear relation, resulting in the following expression:

$$a = g(Wx + b) \tag{2.1}$$

The fact that the information moves along the network in one direction only and without feedback is what gives the name feedforward to these systems (Figure 2.1).

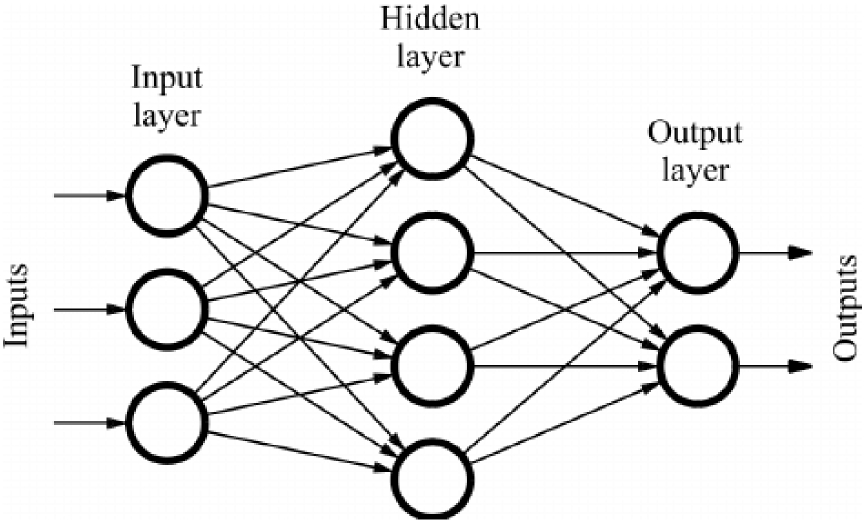


FIGURE 2.1: A feedforward neural network schematic.

Thus, a feedforward network has no notion of order on neither temporal nor spatial dimension, and the only input it considers is the current example it has been exposed to. Feedforward networks are amnesiacs regarding their past; they do not have any type of memorization mechanism and can only remember the specific examples of training.

2.2 Recurrent neural networks

Recurrent neural networks, also known as RNNs, are a class of neural networks that allow previous outputs to be used as inputs while having hidden states contrary to what happened with feedforward networks. Concretely, they are implemented with loops or connections between units allowing a shared-propagation of information from one timestep of the network to the next one, taking into account for computing a new state the previous one (Figure 2.2). Mathematically, each new state is computed as follows:

$$h(t) = g(Wx_t + Uh_{t-1}) \tag{2.2}$$

where t represents the actual timestep, g an activation function, h the state of the network, x the input series, and W, U are learnable weights.

RNNs present a series of advantages that have been explored for time series. As computation takes into account historical information and weights are shared across time, they are particularly well suited for modeling strong-based sequentially data. Also, model size does not increase with the size of the input, which can be a problem when handling series with a high number of input timesteps when using feedforward networks.

However, RNNs are well known for presenting difficulties to access information from a long time, as they do not implement any kind of memory mechanism (known as the *vanishing gradient* problem). Moreover, as it only processes one timestep at a time, when the series presents a high variability or the periodicity is not obvious, it may have adaptative problems.

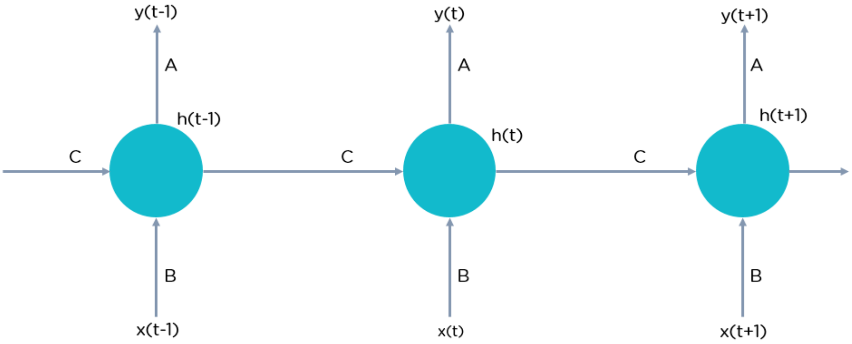


FIGURE 2.2: A recurrent neural network schematic.

As with feedforward networks, RNNs do not present any advantage for the spatial dimension, modeling each zone independently.

2.2.1 Long-short term memory networks

Long short-term memory networks (LSTM) [119] are a variation of the previously presented RNNs which are able to model long-term relations by including a specific memory mechanism: units called memory blocks. Furthermore, another type of unit called gates controls the flow of information from a previous timestep to the next one.

First, the LSTM has to decide which information should be omitted from the cell in the current timestep. To do so, a sigmoid activation function is used. Concretely, it considers the previous state (h_{t-1}) along with the current input x_t and computes the following function:

$$f_t = \sigma(W_f \cdot [x_t, h_{t-1}] + b_f) \tag{2.3}$$

f_t is called the forget gate and resolve which information from previous timesteps should not be remembered. Again, t represents the actual timestep, h the state of the network, x the input series, W represent a learnable weight, and b the bias term.

On the contrary, the so-called input gate (i_t) determines which information to let through based on the current timestep. It writes as:

$$i_t = \sigma(W_i \cdot [x_t, h_{t-1}] + b_i \tag{2.4}$$

From this information, a new cell state (C_t) can be computed by modulating how much from past cell state is kept via the forget gate, and how much new input is allowed via the input gate:

$$C_t = f_t \circ C_{t-1} + i_t \circ tanh(W_c \cdot [x_t, h_{t-1}] + b_c) \tag{2.5}$$

Finally, the new state is calculated:

$$o_t = \sigma(W_o \cdot [x_t, h_{t-1}] + b_o) \tag{2.6}$$

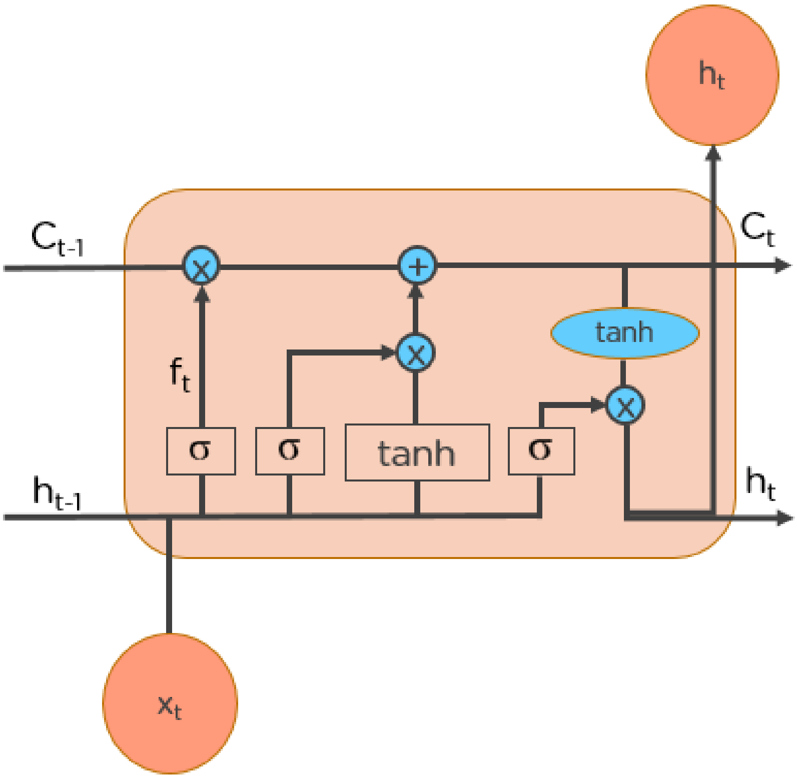


FIGURE 2.3: A LSTM network schematic.

$$h_t = o_t \circ tanh(C_t) \tag{2.7}$$

As with the models seen so far, the spatial dimension is not treated in any special way. A schematic that summarizes this type of network can be found in Figure 2.3.

2.3 Convolutional neural networks

Convolutional Neural Networks (CNNs) [159] are based on the idea of the convolution operation. Convolution itself ($*$ operator) has the following form for 2D images:

$$(x * K)(i,j) = \sum_{m}^{k_1} \sum_{n}^{k_2} x(m,n)K(i-m, j-n) \tag{2.8}$$

where K is the so-called kernel and x the input series which, in this case, it has the form of an image. Thus, CNNs are characterized by learning a series of filters which values depend on how adjacent elements are related. Convolution allows for the encoding of the spatial properties of the input in such a way that propagates the information taking into account spatial relations based on closeness (Figure 2.4). Also, this sharing of information provokes a more efficient type of networks since fewer parameters are needed. CNN filters or kernels, obtained by the convolution of inputs and weights, are local

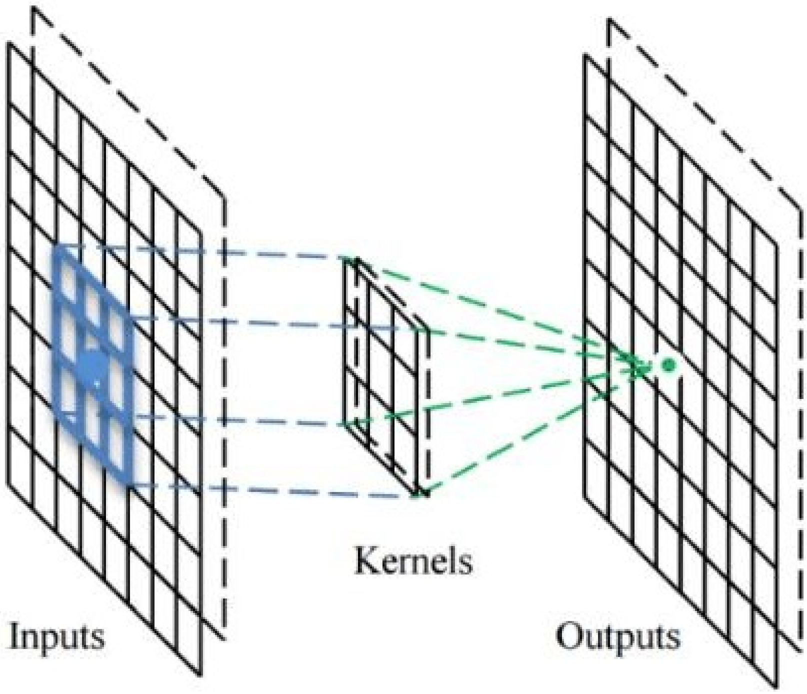


FIGURE 2.4: A convolutional neural network schematic.

in input space and are able to exploit the strong, spatially local correlation present in both dimensions for the spatio-temporal series. Thus, they work well for identifying patterns within local regions of the data which then will be used by subsequent layers to form more complex patterns in a similar way as feedforward networks modeled more complex relations via deepening.

2.4 Attention mechanisms

The fundamental operation of any attention mechanism is based on the self-attention operation [29] (Figure 2.5). It is defined as sequence-to-sequence operation, meaning that the model will have a sequence of elements as input to output another sequence of elements. Given an input vector $x_1, ..., x_t$, and the corresponding output vector $y_1, ..., y_t$, both with dimension k for simplicity, to comùte the output at a given timestep y_i, the self-attention operation simply takes a weighted average over all the input timesteps:

$$y_i = \sum_j w_{ij} x_j \tag{2.9}$$

Where the summation runs through the entire input sequence and the weights sum to one over all j. The weight w_{ij} is not a learnable parameter, as in a normal neural network, but it is derived from a function over x_i and x_j. Called attention weights, w_{ij} are interpreted as how much from input timestep j is being used to compute output timestep i. In other words, the attention mechanism is learning which parts of the input sequence should be

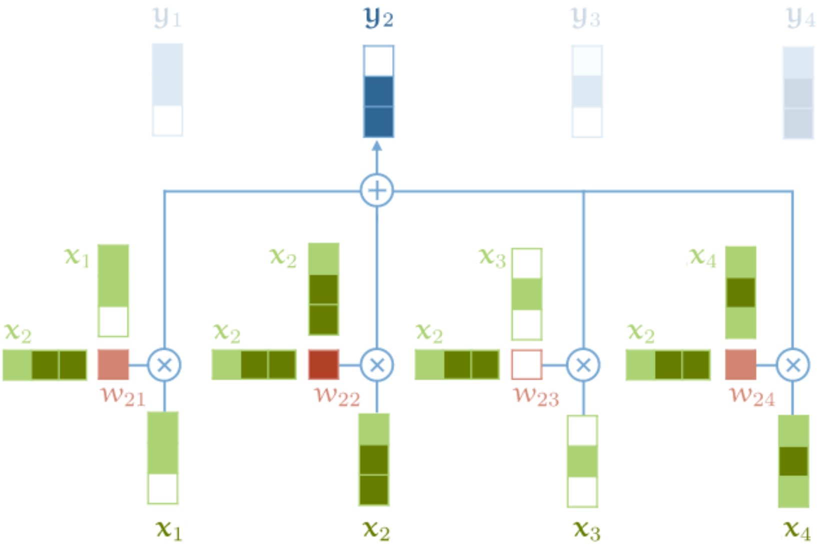


FIGURE 2.5: An attention mechanism schematic.

taken into account for predicting each output timestep. The simplest option for this function is the dot product:

$$w'_{ij} = {x_i}^T x_j \tag{2.10}$$

The dot product is not truncated to any pair of values, so a softmax is applied to map the values to $[0, 1]$ and to ensure that they sum to 1 over the whole sequence:

$$w_{ij} = \frac{\exp w'_{ij}}{\sum_j \exp w'_{ij}} \tag{2.11}$$

Although in principle these mechanisms have been used mainly for sequentialized series only, we will see in Chapter 3 that it is possible to generalize them to pay attention to elements with other characteristics.

2.5 Training neural networks

Training a neural network involves using some training examples that let us update the model weights to create a good mapping of inputs to outputs. This learning process is solved using an optimization algorithm that searches through a space of possible values for a set of weights that results in a good performance on the training dataset via a cost function previously defined. Thus, the objective is to find a set of neural parameters that result in minimum error with respect to this cost function.

This entire process is iterative, meaning that it progresses step by step with small updates to the model weights each iteration and, in turn, a change in the performance of the model each iteration.

However, this optimization problem is considered hard: the error surface is non-convex and contains local minima, flat spots, and is highly multidimensional. In general terms, it is considered that gradient descent algorithms are the best options to address these challenging problems in order to train neural networks.

Concretely, this process is made via the so-called backpropagation algorithm [255], which works by computing the gradient of the loss function with respect to each weight by the chain rule, one layer at a time, iterating backward from the last layer to avoid redundant calculations of intermediate terms in the chain rule. It is worth noting that backpropagation is only an algorithm to compute gradients through a complex and directional structure such as a neural network, but the optimization problem is solved by gradient descent as discussed above.

Thus, and depending on the optimization algorithm chosen, the training will depend on a series of external parameters that will govern the optimization process: for neural networks, some of the most important ones are the learning rate (which modulates the degree the weights are updated), number of epochs (number of iterations through the complete train dataset), and regularizers (which make slight modifications to the learning algorithm such that the model generalizes better).

While the training and optimization of neural networks is a broad and complex field of study *per se*, here there will only be given a few hints of common techniques used throughout the thesis that aim to improve the efficiency and stability of the training performed throughout the different experiments: hyperparametrization methods, learning rate decay, and early stopping.

2.5.1 Hyperparametrization

As we have seen previously, the training of a neural network involves the definition of a series of parameters external to the network itself, which in part will determine the success of the training. These variables, commonly called hyperparameters, need to be tuned externally. Therefore, several methods have been developed to automate this search in such a way that neither a great expert knowledge in the field nor an excessive amount of resources and time are necessary.

All these methods are based on validating the model with different configurations of hyperparameters to find the permutation that minimizes a cost function, in a similar way to the process for training. In particular, the most important methods are the following ones:

- **Grid search:** is simply an exhaustive searching through a manually specified subset of the hyperparameter space of the neural network. Since the parameter space of the algorithm may include real-valued or unbounded value spaces for certain hyperparameters, manually set bounds may be necessary before applying this methodology.

- **Random search:** replaces the exhaustive enumeration of all combinations explained above by selecting them randomly. It can outperform the previous method, especially when only a small number of hyperparameters affects the final performance of the neural network training.

- **Bayesian hyperparametrization:** is defined as building a probability model of the objective function and using it to select the most promising hyperparameters to evaluate in the true objective function. This probability model, which is usually called a *surrogate model*, is represented as $P(error|hyperparametercombination)$, and is repeatedly updated using new information from previous steps. Unlike grid search and random search, the Bayesian approach keeps track of past evaluation results and has a less efficient computational evaluation, but as it incorporate new knowledge over the process, less iterations are needed in comparison.

2.5.2 Learning rate decay

When using stochastic gradient descend, the value for learning rate can be left as system default or can be selected through a wide variety of techniques (for example, those ones explained in Section 2.5.1). Among these techniques, learning rate schedule is based on changing the learning rate during training, usually between epochs/iterations. One of the main ways of proceeding is the so-called decay [276].

This method consists of defining a high initial learning rate whose value will decrease as the training progresses. This decay generally takes one of the following forms: time-based, step-based and exponential.

Decay serves to drive the learning to a soft minimum avoiding oscillations, a situation that usually arises when a too high constant learning rate makes the learning jump back and forth over an optimum point. This idea is consistent with the nature of training if one considers the phase space of the cost function, allowing to speed up the process: at the beginning, the initial weight configuration is expected to be far from the minimum, so making more aggressive updates helps to converge faster. However, as learning progresses, it is closer to the minimum and therefore it is desirable to better refine the subsequent steps. Also, an initially large learning rate help to avoid the memorization of noisy data while decaying progressively the learning rate improves the learning of complex patterns.

2.5.3 Early stopping

Early stopping [328] is a regularization technique used to avoid overfitting when training a neural network with an iterative method such as gradient descent. Given that those algorithms update the network's weights so they better fit the training data each iteration, it is possible for the network to memorize training examples, causing a drop in performance for the test set. Thus, in general terms, the iteration improves the learner's performance on data outside of the training set. However, there is a point in which improving the network's fit to the training data comes at the expense of increased generalization error. Early stopping prevents this effect by providing guidance of how many iterations can be run before the learner begins to over-fit. By defining a number of epochs in which, if there is no improvement in the validation set, training is stopped, this loss of generalization capacity is avoided.

Chapter 3

A spatio-temporal attention-based spot-forecasting framework for urban traffic prediction

Type:	Published article
Journal:	Applied Soft Computing
Authors:	Rodrigo de Medrano & José Luis Aznarte
Published:	August 2020
Impact factor:	5.472
5-Year Impact factor:	5.390
Quartile:	Q1 (Artificial Intelligence)
DOI:	10.1016/j.asoc.2020.106615
Contribution:	Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing.

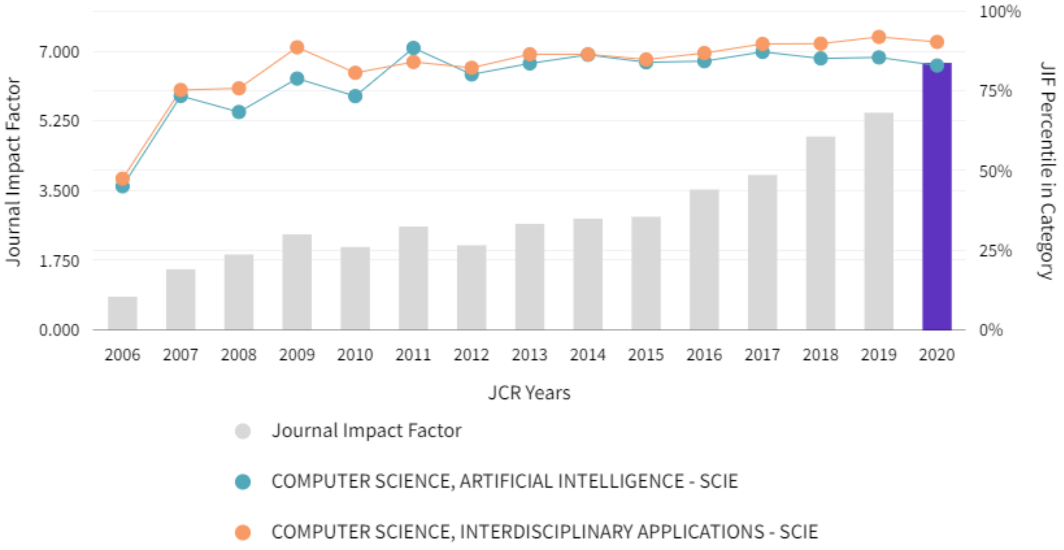


FIGURE 3.1: Impact factor: Applied Soft Computing.

Applied Soft Computing Journal 96 (2020) 106615

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

A spatio-temporal attention-based spot-forecasting framework for urban traffic prediction

Rodrigo de Medrano, José L. Aznarte [*]

Artificial Intelligence Department. Universidad Nacional de Educación a Distancia — UNED, Madrid, 28041, Spain

ARTICLE INFO

Article history:
Received 6 April 2020
Received in revised form 14 July 2020
Accepted 3 August 2020
Available online 8 August 2020

Keywords:
Deep learning
Neural networks
Spatio-temporal series
Traffic forecasting

ABSTRACT

Spatio-temporal forecasting is an open research field whose interest is growing exponentially. In this work we focus on creating a complex deep neural framework for spatio-temporal traffic forecasting with comparatively very good performance and that shows to be adaptable over several spatio-temporal conditions while remaining easy to understand and interpret. Our proposal is based on an interpretable attention-based neural network in which several modules are combined in order to capture key spatio-temporal time series components. Through extensive experimentation, we show how the results of our approach are stable and better than those of other state-of-the-art alternatives.

1. Introduction

Spatio-temporal forecasting is playing a key role in our efforts to understand and model environmental, operational, and social processes of all kinds and their interrelations all over the globe. From climate science and transportation systems to finances and economic, there are plenty of fields in which time and space might constitute two entangled dimensions of data, with one affecting the other and thus both being relevant for prediction. In this context, there is an increasing trend to develop and improve methodologies for gathering and using vast amounts of spatio-temporal data over the last years. Tailored to extract usable knowledge from these big data repositories, there are plenty of proposals trying to facilitate a shared understanding of the multiple relationships between the physical and natural environments and society (being the UE's projects Digital Earth [1] or Galileo [2] two salient examples). By contributing in this direction, it is possible to enrich plenty of services in many ways and gain a better understanding of our world. While machine learning has been widely used for spatio-temporal forecasting in the last decade, there is still room for improvement in our understanding of the models and in their applications.

Specifically, when using neural networks (NN) for regression tasks it is highly desirable that these intelligent systems are capable of adapting to a wide range of circumstances within the framework in which they have been trained. As this ability depends on the data and problem in which the NN is being applied, every field might present different aspects in which it could be beneficial.

In the concrete case of spatio-temporal forecasting, the prediction depends fundamentally on two dimensions: the time horizon and the spatial zone in which the NN is being trained. Thus, traditionally NN are trained and evaluated over some fixed spatial and temporal conditions, restricting the contexts in which they can be applied, making them less suitable to deal with atypical inputs and limiting the knowledge about its general behavior. While creating a system that can infer future properties of the series with a single training is out of the scope with actual techniques, it is important to evaluate algorithms over different spatio-temporal scenarios as every methodology usually presents dissimilar behaviors in distinct situations. Thus, even if a fixed application is intended, exploring the adaptability to different circumstances of an algorithm might be positive.

On the contrary, we propose to characterize spatio-temporal frameworks via a complete and comprehensive experimentation and evaluation over both dimensions. This evaluation methodology, which has been named for convenience as spot-forecasting (in analogy with the economic term *Spot Market*), explores the adaptability of neural systems to any spatio-temporal input for a specific series. Its name refers to the property of these models to predict at any moment in which the forecast is needed. Given some forecasting conditions as number of input–output timesteps and spatial points, the idea is to train and evaluate the network with any possible temporal sequence from the series for a wide range of spatial allocations of different nature. For example, instead of making 24 h prediction starting at 00.00 every day for each point, making 24 h predictions whose start can be any possible hour of the day. Even if a system will work under a rigid

[*] Corresponding author.
E-mail address: jlaznarte@dia.uned.es (J.L. Aznarte).

https://doi.org/10.1016/j.asoc.2020.106615
1568-4946/© 2020 Elsevier B.V. All rights reserved.

scenario, this strategy lets us gain a wider insight of the model, facilitates its application to other spatio-temporal conditions (directly or via transfer learning), makes a more robust model to unusual inputs and works as a data-augmentation technique due to the increase of training population (in the previous example, from one training sample per day to 24 training samples per day).

Pointing in this direction, through this work we propose a novel Neural Network framework called CRANN (from Convo-Recurrent Attentional Neural Network) that is evaluated for several spatio-temporal conditions and compared with some of the state of the art methods. The model presented in this paper is built on the idea of the classical time series decomposition, which attempts to separately model the available knowledge about the underlying unknown generator process. This generator process is usually considered to be composed of several terms like seasonalities, trend, inertia, and spatial relations, plus noise. Thus, our framework is defined like a composition of several modules that exploit different neural architectures in order to separately model these components and aggregate them to make predictions.

Hence, we use a temporal module with a Bahdanau attention mechanism in charge of study seasonality and trend of the series, a spatial module in which we propose a new spatio-temporal attention mechanism to model short-term and spatial relations, and a dense module for retrieving and joining both previous modules together with autoregressive terms and exogenous data in an unique prediction. While we expect spatial and temporal modules to use inertia information too, we reinforce this component with autoregressive terms as deep neural models has shown lack of ability in modeling it (see Section 3.2.3).

Thanks to their capability to provide extra information about the network intra-operation and feature importance, interpretability and explainability are growing in importance and relevance. As we are especially interested in demonstrating that CRANN modules have the behavior just discussed, interpretability is notably useful in our case. Concretely, attention mechanisms are gaining supporters thanks to their capability of achieving good performance, generalizing, and introducing a natural layer of interpretability to the network. Thus, both temporal and spatial modules are covered. The dense module makes use of SHAP values for estimating how important is each component for the final prediction.

In order to showcase the proposed forecasting framework, the problem of traffic intensity prediction is tackled in this paper. This real world problem represents a perfect example of long, high-frequency time series which are spatially interrelated, highly chaotic and with a clear presence of the four aforementioned classical time series components. Furthermore, its environmental, economic, and social importance turns it into a very relevant problem in need of operational and cheap solutions.

In fact, with the increase of vehicles all over the world, several complications have appeared recently: from traffic jams and their impact on economy and air quality, going through traffic accidents, and health-related issues, to name a few. Owing to the relevance of the matter, intelligent transport systems have arised as an important field for the sake of improving traffic management problems and establishing sustainable mobility as a real option. As an immediate consequence, traffic prediction can be considered as a crucial problem on its own and a perfect candidate as a real application that could benefit from adaptable, accurate and interpretable NNs. For example, these kind of systems might help to improve route-recommendation systems by not only estimating but predicting, to optimize in real time buses waiting times, and to extract better spatio-temporal information that would be helpful for traffic planning and management. Although traffic systems are usually focused on short-term forecasting,[1] for academic purposes we tackle the long-term problem by predicting 24 h in order to demonstrate that our model is capable of learning intrinsic spatio-temporal traffic dependencies and patterns. However, as we will show, the model is easily adaptable to any forecast window.

The main contributions of this study are summarized as follows:

- A new deep neural network framework especially designed for spatio-temporal prediction is proposed.
- A novel spatio-temporal attention-based approach for regression is presented.
- The contribution is illustrated by tackling a traffic prediction problem which is considered hard in both dimensions.
- Results show that our proposal beats other state-of-the-art models in accuracy, adaptability, and interpretability.

The rest of the paper is organized as follows: related work is discussed in Section 2, while Section 3 presents the problem formulation and our deep learning model for spatio-temporal regression. Then, in Section 4 we introduce our dataset, experimental design and its properties. Section 5 illustrates the evaluation of the proposed architecture as derived after appropriate experimentation. Finally, in Section 6 we point out future research directions and conclusions.

2. Related work

2.1. Deep neural networks for spatio-temporal regression

Classic statistical approaches and most of the machine learning techniques that are used to deal with spatio-temporal forecasting sometimes perform poorly due to several reasons. Spatio-temporal data usually presents inherent interactions between both spatial and temporal dimensions, which makes the problem more complex and harder to deal with by these methodologies. Also, it is very common to assume that data samples are independently generated but this assumption does not always hold because spatio-temporal data tends to be highly self correlated.

On the contrary, models based on deep learning present two fundamental properties that make them more suitable for spatio-temporal regression: their ability to approximate arbitrarily complex functions and their facility for feature representation learning, which allows for making fewer assumptions and permits the discovery of deeper relations in data.

Within deep learning, almost all types of networks have been tried for spatio-temporal regression. The most common ones are recurrent neural networks (RNN), which due to its recursive structure have a privileged nature for working with ordered sequences as time series. Nevertheless, it is not easy to use them to model spatial relations, which makes them less suitable for this kind of problems. For this reason, RNN models are usually combined with some spatial information, as convolutions or spatial matrices. Previous works within the RNN group are [3,4] for example. While RNNs have received a lot of attention during last years, interest in convolutional neural networks (CNN) for spatio-temporal series is recently growing. Not only these systems are capable of exploiting spatial relations, but they are showing state-of-the-art performance in extracting short-term temporal relations too. For example, [5,6] propose the use of CNN in spatio-temporal regression.

[1] Traffic forecasting is commonly classified as short-term if the prediction horizon is less than 30 min and long-term when it is over 30 min. We adopt that terminology throughout this paper.

In recent years, more complex models based on both RNN and CNN are replacing traditional neural networks in this kind of problems. This is the case of sequence to sequence models (seq2seq) and encoder–decoder architectures. By enlarging the input information into a latent space and correctly decoding it, these models have induced a boost in spatio-temporal series regression. As it happened with RNN, spatial information is usually introduced explicitly. Some examples might be found in [7,8]. Finally, attention mechanisms were introduced by [9,10] for natural language processing. However, some researches have recently shown their ability to handle all kinds of sequenced problems, as time and spatio-temporal series. Particularly, they have demonstrated to be a promising approach in capturing the correlations between inputs and outputs while including a natural layer of interpretability to neural models. These attention mechanisms might be introduced at any dimension: spatial [11], temporal [12] or both of them [13].

For a survey that recapitulates the main characteristics of deep learning methods for spatio-temporal regression and a vast compilation of previous work, see [14].

2.2. *Traffic prediction*

Traffic flow prediction has been attempted for decades, and has experienced a strong recent change after the emerging methodologies that let us model different traffic characteristics. With the increase of real-time traffic data collection methods, data-based approaches that use historical data to capture spatio-temporal traffic patterns are every day more common. We will divide this data-driven methods into three major categories: statistical models, general machine learning models, and deep learning models.

Within statistical methods, the most successful approach has been ARIMA and its derivates, which have been used for short-term traffic flow prediction [15]. Afterwards, some expansions as SARIMAs [16] were also proposed to improve traffic prediction performance. Nevertheless, these models are constrained according to several assumptions that, in real world data as our, do not always fit properly.

Among general machine learning approaches, Bayesian methods have shown to adapt well when dealing with spatio-temporal problems [17,18], as their graph structure fits in a road-network visualization. However, they do not always show better performance when compared to other methodologies that will be presented next. Another usual technique in the field of forecasting spatio-temporal series are tree models. Within this area, there are different approaches [19,20], each one with its own advantages and disadvantages. Generally tree models are easily interpretable, making them a good option if the main interest is to better understand the phenomena. Nevertheless, tree models tend to overfit when the amount of data and dimensions of the problem is big, as it normally is the case in traffic prediction. As with trees, support vector machines (SVM) and support vector regression (SVR) have been widely used, as in [21,22]. While SVM and SVR perform well, these methodologies must establish a kernel as a basis for constructing the model. This means that, for such a specific problem like the one we are working on, the use of a predetermined kernel (usually radial) might not be flexible enough. Bioinspired techniques, although less used, show a promising ability to optimize traffic-related processes. Within these processes, traffic prediction is usually one of the main steps. For example, [23] illustrates a rerouting system through a pheromone model capable of estimating future traffic states to reduce traffic congestion.

In a closer line with our work, deep learning has been widely used for traffic forecasting. The idea of stacking CNNs modules over LSTMs (or vice versa) is usual in recent literature. Some of the most interesting work in this category applied to traffic prediction can be found in [24,25]. Furthermore, [26] shows that the combination of these modules together with an attention mechanism for both space and time dimensions, might be beneficial. In this same line, [27] proposes a spatio-temporal attention mechanism and show how through interpretability we can extract valuable information for traffic management systems. Lately, other options have been considered as using 3 dimensional CNNs for making the predictions [28] to effectively extract features from both spatial and temporal dimensions or combining CNN–LSTM modules with data reduction techniques in order to boost performance [29]. In [30] authors present an example of how to deal with incomplete data while still being capable of exploring spatio-temporal traffic relations. As it was mentioned before, the vast majority of these works have a set of fixed conditions and mainly focus on short-term predictions. Longer-term predictions (with horizons of more than fours hours) can also be found in [31] in which a neural predictor is used to mine the potential relationship between traffic flow data and a combination of key contextual factors for daily forecasting, and [32] where ConvLSTM units try to capture the general spatio-temporal traffic dependencies and the periodic traffic pattern in order to forecast one week ahead.

In concordance with these last works, our model is designed to be adaptable to both long and short term forecasting. Also, it is not limited nor evaluated over a set of fixed conditions, letting us extract more general conclusions.

2.3. *Time series decomposition in deep neural models*

Time series decomposition and derived methods for regression have been widely studied in the statistical context. Beyond standard methodologies, as ARIMA and exponential smoothing, more elaborated proposals have been suggested. For example, in [33] a bootstrap of the remainder for bagging several time series via exponential smoothing is proposed. Similarly, [34] presents an extension of an analogous methodology using SARIMA. In both cases, it is demonstrated that proper use of time series components for modeling can be profitable and thus this remains as a promising research line.

In the concrete case of deep neural models, although using time series decomposition in order to improve and boost the performance of deep neural networks is not new, most of previous research has focused on using those components externally to the network. Several studies point out that, before feeding the network, it might be beneficial to detrend the series to just build a prediction model for the residual series [35,36]. Other works show how autoregressive methods together with deep neural models help to tackle the scale insensitive problem of artificial neural networks [37] and allow for the implementation of several temporal window sizes for training efficiently [38]. Deseasonalisation in order to minimize the complexity of the original time series has been recommended through several works too [39,40]. For concrete spatio-temporal series, [41] shows that time series residuals not only represent random noise, but also can capture spatial patterns, making working with time series components even more interesting

However, the way in which neural networks relate to time series components remains an open issue. Although it has been demonstrated that a correct decomposition of the series can help the system, it is not clear how deep neural models can deal with these components by themselves without the need of external information as we propose.

2.4. Interpretability

Defined as the ability to explain or to present in understandable terms aspects of a machine learning algorithm operation to humans, interpretability is growing in importance, especially in deep neural models due to its black-box nature. Until now, it has principally been investigated and demonstrated in a wide variety of tasks such as natural language processing, classification explanation, image captioning, etc [10,42,43]. However, in regression problems and particularly in traffic is still an open issue and there is a long way to go. For example, [44] demonstrates that by using a bidirectional LSTM that models paths in the road network and analyzing features from the hidden layer outputs it is possible to extract important information about the road network. Similarly, [45] studied the importance of the different road segments when forecasting traffic via a graph convolutional-LSTM network. In general, traffic interpretability research has focused in pointing out important road segments. On the contrary, [27] presents a comprehensive example of spatio-temporal attention in which both dimensions are analyzed from an interpretability point of view, and not only from the spatial one.

Following this last idea, we propose a methodology in which spatio-temporal interpretability is taken into account and, at the same time, go deeper in understanding how important these two dimensions are to model the generator process of our problem.

3. CRANN model and problem formulation

3.1. Problem formulation

Given a spatial zone S, where each traffic sensor is represented as s_i, and a timestep t_j, we aim to learn a model to predict the volume of traffic in each sensor s_i during each time slot t_j. This mean that a spatio-temporal sample writes as $x_{s_i,t_j} : j = 1, \dots, T; i = 1, \dots, S$. From now on, we will distinguish a prediction from a real sample by using $\tilde{x}_{s_i,t_j}$ for the first one.

3.2. CRANN: a combination approach for spatio-temporal regression

As stated above, CRANN model is based on the idea of combining neural modules with the intention to exploit the various components that can be identified in a spatio-temporal series: seasonality, trend, inertia, and spatial relations. By combining different neural architectures focused on each component, we expect to avoid redundant information flowing through the network and to maximize the benefits of each approach. As a result, several layers of interpretability will allow us to better understand the problem being modeled and to verify that our model is working the way we were expecting.

The code for the software used in this paper can be found in https://github.com/rdemedrano/crann_traffic.

3.2.1. Temporal module

There is a consensus that dealing with long-term sequences using ordinary encoder–decoder architecture is a promising approach. However, the fact that only the final state of the encoder is available to the decoder limits these models when trying to make short or long-term predictions [46]. Particularly, in traffic we would expect an improvement in performance when taking into account not only closer states of the desired output, but also several days.

In order to solve this problem, several encoder–decoder architectures that use information from some or all timesteps have been proposed. Among all these models, a particular approach has shown good qualities by improving performance and adding an interpretability layer to the system: attention mechanisms. Presented in several ways [9,10], the idea behind these mechanisms lies in creating a unique mapping between each time step of the decoder output to all the encoder hidden states. This means that for each output of the decoder, it can access the entire input sequence and can selectively pick out specific elements from that sequence to produce the output. In other words, for each output, the network learns to pay attention at those past timesteps (inputs) that might have had a greater impact on the prediction. Typically, these mechanisms are exemplified by thinking of manual translation: instead of translating word by word, context matters and it is better to focus more on specific past words or phrases to translate the next.

Following that rationale, our temporal module is formed by two LSTMs working as encoder and decoder respectively. The first one inputs the time series and outputs a hidden state s of typically higher dimension than the input, while the second one inputs a concatenation of attention mechanism output c (named 'context vector') and the previous decoder outputs, and uses this information to perform its prediction.

As it can be seen, the structure is very similar to a sequence to sequence model without bottleneck but with the introduction of new information via the attention mechanism. The idea behind this model is explained below. For simplicity, notation is coherent with the one used by Bahdanau [10] through this section. For each forecast step i, the context vector is calculated taking into account the encoder hidden state for each input timestep j:

$$c_i = \sum_{j=1}^{N} \alpha_{i,j} h_j, \tag{1}$$

where N is the input sequence dimension (coincident with number of encoder hidden states), $\alpha_{i,j}$ is the attention weight defined as how much from the encoder hidden state j should be payed attention to when making the prediction at time i. It is computed as follows:

$$\alpha_{i,j} = \frac{f(h_i, s_j)}{\sum_{j=1}^{N} f(h_i, s_j)}. \tag{2}$$

In this last expression, h is the decoder hidden state and f refers to an attention function that estimates attention scores between s and h. Depending on the attention mechanism, many functions have been suggested as attention functions (for example, dot products, concatenation, general...). In this work, a feedforward neural network that combines information from both the encoder and the decoder is chosen. Specifically, it writes:

$$f(h_i, s_j) = W_c \cdot tanh(W_d \cdot h_i + W_e \cdot s_j), \tag{3}$$

where Ws are weight matrices.

Finally, the new decoder hidden state h'_i is obtained through concatenating c_i with h_i and the output can be decoded as

$$h'_i = [c_i; h_i]. \tag{4}$$

The complete process is summarized in Fig. 1. The temporal module of CRANN focuses its effort on discovering and modeling long-term time relations of the complete system by using average traffic for the complete zone. This means that we assume that all spatial locations behaves similarly respect to time. In the concrete case of traffic, this assumption usually holds as the temporal distribution for all zones share a common pattern. From Eqs. (1)–(4), it should be clear that no spatial relations have been explored or introduced. Although it might seem more profitable to capture these relations for each spatial series, we would be learning redundant knowledge once that the spatial module comes out. By taking into account information from several past weeks, the model will be capable of capturing the periodicity and trend changes in the series, which might be fundamental for more

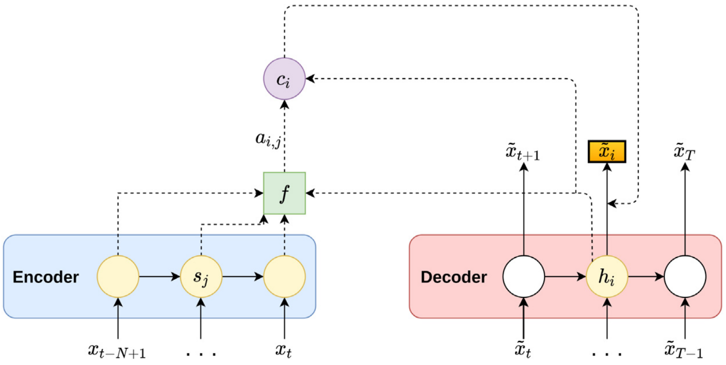


Fig. 1. Schematics of the attention mechanism used in our temporal module when predicting timestep i for a T horizon forecasting. In yellow, hidden states for both encoder and decoder that are used to compute the prediction. In green, the attention function. In violet, the obtained context vector based on attention weights $\alpha_{i,j}$ and h_i. In orange, the predicted value computed by concatenating c_i with h_i.

precise forecasting. As traffic trend is not exactly equal for a temporal window of several hours or days, it is necessary to adapt CRANN temporal module input to an amount of time that let us avoid temporal information loss. In particular, we will use two weeks as input for a 24 h output.

3.2.2. Spatial module

Even though traffic seems highly dependent on its temporal dimension, it is also clear that spatial relations are relevant. The premise of spatio-temporal forecasting is based on not only taking into account that these relations exist, but effectively using and learning them to improve performance. In this context, convolutional neural networks (CNN) appear like a perfect choice as they are meant to precisely exploit spatial characteristics and interactions. Furthermore, as it was mentioned in Section 2, CNNs are also gaining attention as a promising paradigm to study short-term temporal associations. Hence, we propose a novel spatio-temporal attention mechanism that tackles two major aspects of spatio-temporal forecasting with CNN: adds a new layer in order to improve spatial relations and our understanding over them in a specific problem, and lets the CNN explore further short-term temporal information.

As in the temporal case, this mechanism can be introduced as a new layer through the network, meaning that our system will consist of an usual CNN followed by the spatio-temporal attention mechanism. The CNN will enrich input information and compute some output x_{conv} with the same dimensions as its input. In other words, it will be the one in charge to improve the quality of the input while making sure to keep some aspects of the original structure of the series. In some way, it is equivalent to the encoder model from Section 3.2.1, except that there is not an equivalent decoder structure as the own attention mechanism can handle it. This mechanism works by assigning a score $\sigma_{i,j,k}$ to every pair of spatial points (j, k) for each input lag i. The score $\sigma_{i,j,k}$ represents how important is the point k in lag of the input i in order to calculate the prediction for point j for all output timesteps. It writes as:

$$\sigma = g(x_{\text{conv}}, W_{\text{att}}),\ \ W_{\text{att}} \in \mathbb{R}^{T\times S\times S}, \tag{5}$$

where T is the number of timesteps, S the number of spatial points, g is an attention function that calculates an attention score and W_{att} defines the spatio-temporal attention tensor. W_{att} is a learnable tensor which can be interpreted as a means to modulate spatio-temporal interdependencies of the system. It can be decomposed in a three-dimensional space, meaning that $W_{\text{att}}^{i,j,k}$ encodes how does the point j at timestep i interact with the CNN output x_{conv} to make the prediction.

Given that each element of W_{att} is expected to provide information about the system dynamics, the attention function g is useful to modulate concrete relations, element by element, for a given input series. Thus, it is defined as follows:

$$g(x_{\text{conv}}, W_{\text{att}}) = x_{\text{conv}} \circ W_{\text{att}}, \tag{6}$$

where $\circ$ is the Hadamard product (also known as the element-wise, entrywise, or Schur product). Although some other functions as concatenation and a feedforward neural network have been tested, no improvement was reported. Moreover, Hadamard product stands out for its simplicity and for offering a naive explanation about the inner functioning of the spatio-temporal attention mechanism without need of extra learnable parameters.

Once that these attention scores have been calculated, it is preferable to give them some properties that ease the interpretation and convergence of the attention mechanism. The spatio-temporal attention matrix $a \in \mathbb{R}^{T\times S\times S}$ is defined as a three-dimensional tensor that meets the following conditions:

- Each attention weight is constrained between zero and one: $a_{i,j,k} \in [0, 1]$.
- As $a_{i,j,k}$ represents the importance of point k at timestep i to predict j, the sum of attention weights for each timestep i and point j must add up to one: $\sum_{k=1}^{S} a_{i,j,k} = 1$.

By enforcing these conditions we can therefore infer a probabilistic interpretation of the attention weights. This can be done by applying a softmax operator over the third dimension:

$$a = \text{softmax}(\sigma). \tag{7}$$

Finally, in order to calculate the definitive prediction $\tilde{x}$, we use the inner product between tensors. This way we can easily interpret the output as a weighted sum over all spatio-temporal input conditions through the attention weights. Depending on how relevant each input element is for the regression, it will contribute differently to the final output:

$$\tilde{x} = a \cdot x_{\text{conv}}. \tag{8}$$

The complete process is summarized in Fig. 2. To conclude, the spatial module of CRANN focuses its efforts on discovering and modeling spatial and short-term time relations of the complete system by using real traffic data for each sensor. Given that the spatial module focus on modeling short-term patterns, we avoid learning the same information as the temporal module. Furthermore, spatial relations in traffic are mainly important in the short-term, meaning that this dimension would not especially benefit from long-term information. For the CNN, a 2D model

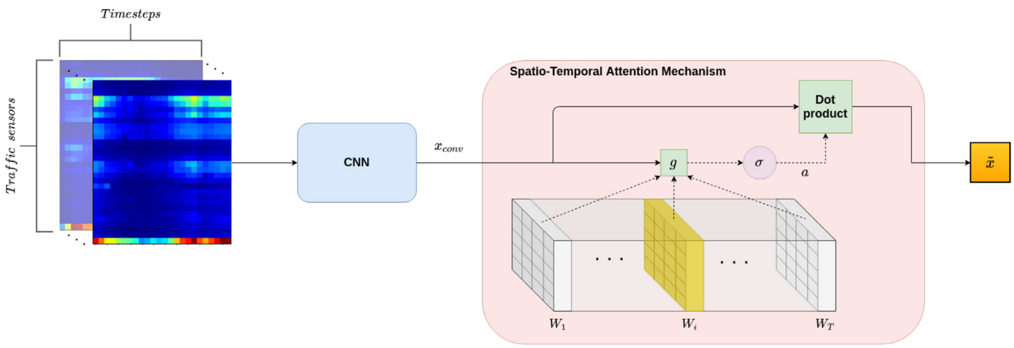


Fig. 2. Schematics of the attention mechanism used in our spatial module when predicting all timesteps for a T horizon forecasting given a batch of inputs. In yellow, a representation of W_{att} for an input timestep i. In green, the attention function and dot product. In violet, the obtained score vector σ based on x_{conv} and W_{att}. In orange, the predicted value.

in which every channel corresponds to a timestep is used. The architecture consists of a sequence of convolutions layers, batch normalization, and ReLU activation. In particular, we will use a 24 h input for a 24 h output.

3.2.3. *Dense module and training*

At this point, on the one hand we have modeled the general behavior of all the involved time series and we thus have trend information from the temporal module, and on the other hand we have explored spatial relations and specific predictions for each traffic sensor through the spatial model. Hence, it is necessary to join both modules in some way that let us exploit all the available information for the sake of improving the final performance. At the same time, it might be interesting to introduce available exogenous knowledge that might affect the future of the series. While several exogenous variables are well known as important for traffic forecasting, we will only use meteorological features as we reckon they might be enough to prove how our model works when using exogenous data in a first approximation. Lastly, since inertia has a central role in time series forecasting, we include the timesteps $t-1$ to $t-4$ as autoregressive terms. Although we might expect that both previous modules take into account this inertia at some level, as Ling et al. [37] pointed out, *due to the non-linear nature of the convolutional and recurrent components, one major drawback of the neural network model is that the scale of outputs is not sensitive to the scale of inputs*, meaning that in real datasets with severe scale changing like ours this effect might be problematic. Thus, making use of this information directly is also expected to benefit the final performance.

The resulting CRANN architecture is shown in Fig. 3. It consists of a dense module whose inputs are the exogenous data, the autoregressive data and the outputs of both the temporal and spatial previously described modules. This last dense module is simply a fully connected feedforward neural network that modulates all previous information before making the final prediction. As all spatial and temporal information have been already considered and managed by the rest of the modules, a simple feedforward network can handle this final part satisfactorily.

While all these modules could be stacked, we decided to use a mixed parallel/series structure for the sake of improving modularity and explainability through the network. By having a compendium of models with a specific and clear job working independently, it is easier to train, improve, remodel, or change any of them if needed. Moreover, the stacked approach was tested but no significant accuracy improvement was reported.

Regarding training, it can be done at once or in several steps (for each module) in order to parallelize the process. Also, all weights are randomly instantiated using "Xavier" initialization. Finally, and independently on how the network is being trained, the training can be summarized as with any other neural network as searching some parameters θ^* via minimization of a cost function L. Concretely, our cost function will be the commonly used Mean Squared Error (MSE). The process can be summarized as:

$$\begin{aligned} \theta^* &= \arg\min_{\theta} L(\theta), \\ L(\cdot) &= \frac{1}{TS}\sum_{j=1}^{T}\sum_{i=1}^{S}(\tilde{x}_{s_i,t_j} - x_{s_i,t_j})^2 \end{aligned} \tag{9}$$

3.2.4. *Interpretability*

The ability to interpret the trained models is nowadays a must-have in every machine learning research check-list. For that reason, we should value methodologies that are able to offer explanations about their predictions. In the particular case of CRANN, interpretability has been put into practice as follows:

- Temporal module: By using a temporal attention mechanism, we have an intrinsic interpretability layer. Since we defined attention weights as how important each lag from the input sequence is for predicting each output timestep (see Section 3.2.1 for a deeper insight), we can easily interpret these weights to better understand how is our temporal module making use of the inputs when forecasting.
- Spatial module: As with the temporal module, the underlying attention mechanism provide an easy and natural interpretation. Attention weights typify how significant is every spatial point when predicting in the spatio-temporal domain. Furthermore, they might be represented by input lag or aggregated.
- Dense module: As the information flowing through the previous modules has a clear interpretation (the temporal module outputs the average traffic for the whole space and the spatial module outputs actual spatio-temporal predictions), it is straightforward to interpret the network with several feature analysis methods (like integrated gradients [47]) or saliency methods (like SHAP values [48]). In this work, SHAP values are chosen. It is important to remark that with a non-parallel join of modules (Section 3.2.4), these methodologies might not be as convenient due to non-explainable inputs of the dense module, i.e., by having interpretable middle stages through the network is easier to elucidate if certain information is contributing to the final prediction or not.

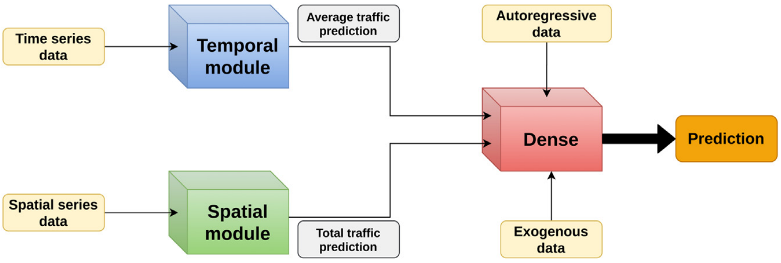


Fig. 3. Schematics for the CRANN architecture. In yellow, input data sources. Time series data refers to average traffic historic data, spatial series data to the real traffic historic, and exogenous data are in this case weather predictions. In gray, output from the different middle stages. In orange, the final spatio-temporal prediction.

4. Data and experiments

To characterize and validate our proposed model, this section provides information related to all the decisions taken and the experiments performed. As explained above, we focus our work on the long-term forecasting problem, i.e., a 24 h spatio-temporal prediction.

4.1. Data description and analysis

To validate the CRANN framework for spatio-temporal forecasting, we chose the problem of predicting traffic intensity in the city of Madrid. The data available came from two different sources:

- **Traffic data:** Provided by the Municipality of Madrid through its open data portal,[2] this dataset contains historical data of traffic measurements in the city of Madrid. The measurements are taken every 15 min at each point, including traffic intensity in number of cars per hour. Spatial information is given by traffic sensors with their coordinates (longitude and latitude). While a dense and populated network of over 4.000 sensors are available, we decided to simplify and use only a selection of them, as explained below.
- **Weather data:** Weather data was also provided by the Municipality of Madrid.[2] Weather observations consist of hourly temperature in Celsius degrees, solar radiation in W/m^2, wind speed measured in ms^{-1}, daily rainfall in mm h^{-1}, pressure in mbar, and degree of humidity in percentage records. Weather information is reported hourly and they are used as if they were numerical weather predictions (feeding the model at each moment with the data corresponding to the forecasting horizon).

In this work, only data from 2018 and 2019 is used.

For a more robust evaluation of the different models, four specific zones are chosen (see Table 1), each one of them containing 30 traffic sensors (Fig. 4). All these four zones are characteristic for being hot spots of traffic in Madrid. In addition, they all present a wide variety of traffic conditions: one-way streets, avenues, highways, roundabouts and, in general, ways with different flow conditions. Statistics presented in Table 1 for each zone point in this direction. Although these spatial dispositions result in a more complicated environment, makes our work more general.

Table 1
Location of the center of spatial zones and name correspondence from now on. Main data statistics.

Zone	Name	Longitude	Latitude	Mean	Std
A	Legazpi	−3.6952	40.3911	563.6	803.2
B	Atocha	−3.6920	40.4087	680.9	769.7
C	Avenida de América	−3.6774	40.4374	459.3	476.6
D	Plaza Elíptica	−3.7176	40.3852	360.2	517.8

Missing values are scarce (about 1% per series). They are replaced by sensor, hour, and day of the week aggregation as interpolation and closeness replacement leads to greater loss of information. Outliers represent less than 0.001% of each series and are given by public events (for example, Champions League final or Basketball World Cup). As these kinds of events are not representative of our problem, and thus they are excluded from our analysis.

The data are aggregated into 1-h intervals and, due to the lack of outliers, normalized using a min–max technique to the range [0,1]. Normalization constants are calculated over the training dataset. Each spatio-temporal series is normalized separately as we are looking for an agnostic scale for each sensor.

In order to better understand our problem, we show significant properties of the data. Due to high number of sensors and the spatial heterogeneity commented above, instead of showing general attributes from our series (as mean, median, or dispersion) it is more instructive to see both spatial and temporal distributions. Thus, on one hand, Fig. 5 shows a boxplot for different time variables. From this figure, it should be clear that traffic is highly dependent on time and periods of human activity. On the other hand, the spatial distribution of our series is displayed in Fig. 6. This last figure not only let us better understand our data, but also reinforces the idea of having very diverse spatial zones for our study, and a great heterogeneity. This secures our experimentation with respect to the variety of data and situations present in this work.

4.2. Benchmark models

We compare the performance of the proposed CRANN with a CNN, an LSTM, the usual combination CNN+LSTM, and a sequence to sequence model (seq2seq).

- CNN: A 2D convolutional model in which every channel corresponds to a timestep. The architecture consists of a sequence of convolutional layers, batch normalization, and ReLU activation. For every layer a kernel size of 3 for each dimension is used. It uses 24 h as input and outputs a 24 h prediction.
- LSTM: These models have several hidden layers with a number of hidden units to determine. We used the tanh activation functions as in the original model. The number of inputs and outputs are equivalent to the number of sensors. Although GRUs modules have also been tested, no difference has been reported. It uses historical data from two weeks as input and outputs a 24-h prediction.

[2] *Portal de datos abiertos del Ayuntamiento de Madrid*: https://datos.madrid.es/portal/site/egob/.

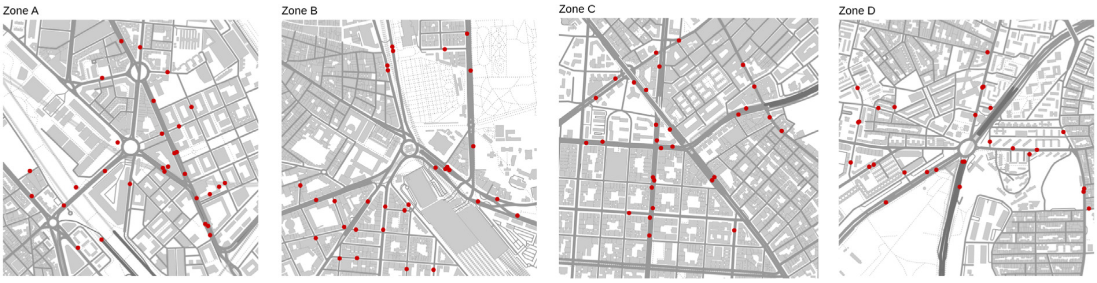


Fig. 4. Location of traffic sensors for each zone.

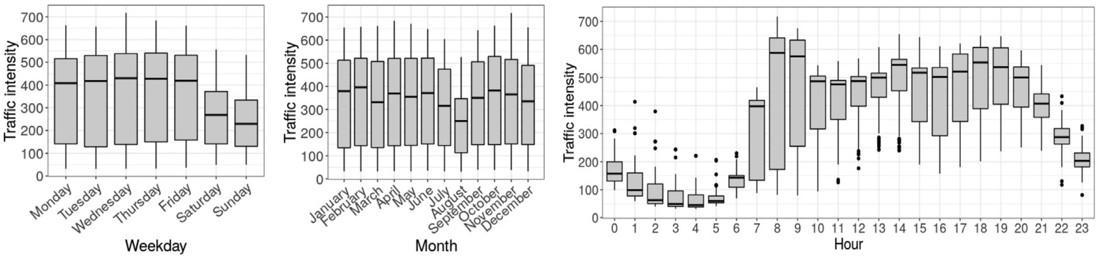


Fig. 5. Monthly, weekly, and hourly distribution of traffic intensity series.

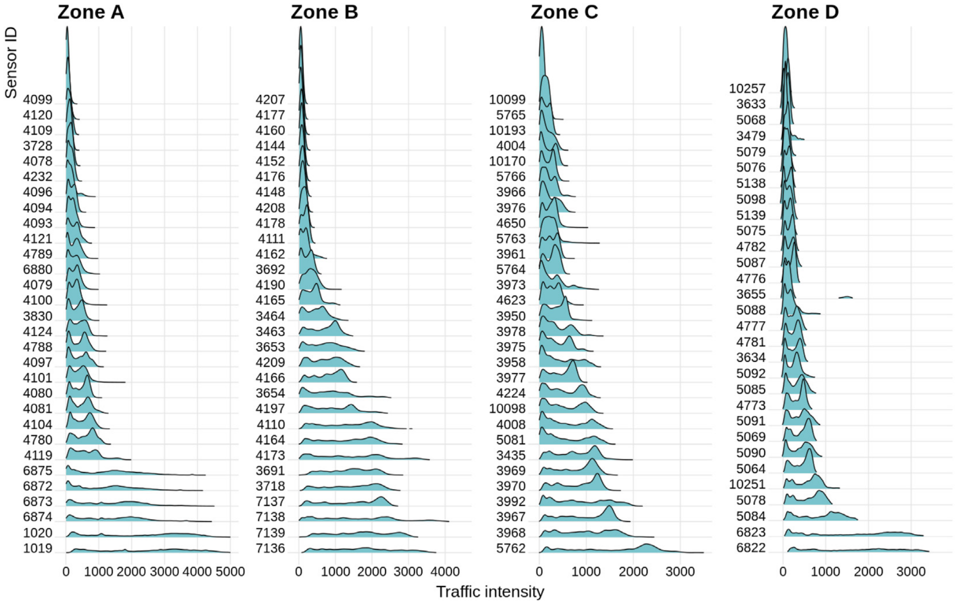


Fig. 6. Traffic intensity distribution by sensor (in number of vehicles per hour).

- CNN+LSTM: A stacked model consisting of a CNN module whose output is in turn the input of a LSTM. Both modules are defined as the two previous models. It uses 24 h as input and outputs a 24-h prediction.
- seq2seq: These architectures are based on an encoder and a decoder, both LSTMs, without "bottleneck". That is to say, hidden variables from all timesteps are used as inputs for the decoder. The number of inputs and outputs are equivalent to the number of sensors. As with LSTMs, GRUs did not show a better performance. It uses two weeks as input and outputs a 24 h prediction.

With these models and their implementation particularities (inputs and outputs) we aim to cover a wide range of neural network paradigms for our comparison. For instance, CNNs are especially designed for learning spatial relations, LSTMs and Seq2Seq models are designed to explore mainly time interactions and CNN+LSTM are closer to our model being a mixture of both previous approaches. Given that the latest evidence points out neural networks based models as the most appropriates techniques for spatio-temporal traffic forecasting, we have only chosen methodologies within this category.

Concerning hyperparametrization and training, instead of using preset architectures, to be fair, the optimal configuration

Table 2
Values used for each hyperparameter and total number of parameters.

Model	Hyperparameter	Value	# parameters
CNN	Convolutions	(32,32,32,64,64,64)	132k
LSTM	Number of layers Hidden units	2 100	206k
CNN+LSTM	Convolutions Number of layers Hidden units	(32,32,64,64,64) 2 100	329k
seq2seq	Number of layers Hidden units	2 100	368k
CRANN	Convolutions Number of layers Hidden units Dense layers	(64,64,64,64,64) 1 100 1	1M

for each model was obtained via Bayesian hyperparameter optimization which is defined as: building a probability model of the objective function and using it to select the most promising hyperparameters to evaluate in the true objective function. This probability model, which is usually called a *surrogate model*, is represented as $P(error|hyperparametercombination)$, and is repeatedly updated using new information from previous steps. Unlike grid search and random search, the Bayesian approach keeps track of past evaluation results and has a less efficient computational evaluation, but as it incorporate new knowledge over the process, less iterations are needed in comparison. Final hyperparameters can be found in Table 2, and they purely depend on this Bayesian optimization process. Although CRANN can train each module separately, the hyperparametrization process must be done jointly to obtain the best hyperparameters when the three modules work together.

All the models are trained using the mean squared error (MSE) as objective function with the Adam optimizer. The batch size is 64, the initial learning rate is 0.01, and both early stopping and learning rate decay are implemented in order to avoid overfitting and improve performance. The experiments run in an Intel Core i7 processor, 32 GB RAM and NVIDIA RTX 2070 GPU. All the code is built over the *PyTorch* package.

4.3. Experimental design

In order to guarantee that models can be compared fairly, it is essential to fix the approach to error estimation, which must be shared as much as possible by all models. First of all, as stated in [49], standard k-cross-validation is the way to go when validating neural networks for time series if several conditions are met. Specifically, that we are modeling a stationary nonlinear process, that we can ensure that the leave-one-out estimation is a consistent estimator for our predictions and that we have serially uncorrelated errors.

While the first and the third conditions are trivially fulfilled for our problem, the second one needs to be specifically studied for the sake of avoiding data leakage. Given that we use all the possible series, even though the ones are unrepeated, it is possible to introduce prior information from the training to the test via closeness of samples (for example training a sequence whose start is at 10:00 AM and testing in a sequence whose start is at 11:00 AM from the same day i.e. one timestep forward). Due to this problem, it is not possible to create random folds and it is necessary to specify a separation border among different sets (training, validation, and test).

In this concrete case, this separation takes as many timesteps as every model uses for its training. A scheme of this methodology is shown in Fig. 7. Particularly, a 10-cross-validation strategy without repetition is used for each spatial zone separately, with a 80%/10%/10% scheme for train/validation/ test sets for each fold. Given that this approach lets all data to be tested, models are validated over the entire spectrum of possibilities for a particular problem and we can assure that the three datasets cover the same space.

Table 3
Average performance for $t = 1$ to $t = 24$, calculated over all spatial zones and average run time per fold. For a more detailed view of error metrics distribution, see Fig. 8.

Model	RMSE	|bias|	WMAPE	Run time (s)
CNN	238.24	22.12	25.89	68
LSTM	255.76	19.58	27.46	552
CNN+LSTM	252.34	21.70	27.29	144
Seq2Seq	246.45	19.14	25.79	1098
CRANN	**221.31**	**17.80**	**23.18**	1083

To evaluate the precision of each model, we computed root mean squared error (RMSE), bias, and weighted mean absolute percentage error (WMAPE). In a spatio-temporal context, they are defined as:

$$\text{RMSE} = \sqrt{\frac{1}{TS}\sum_{j=1}^{T}\sum_{i=1}^{S}(\tilde{x}_{s_i,t_j} - x_{s_i,t_j})^2}, \tag{10}$$

$$\text{bias} = \frac{1}{TS}\sum_{j=1}^{T}\sum_{i=1}^{S}(\tilde{x}_{s_i,t_j} - x_{s_i,t_j}), \tag{11}$$

$$\text{WMAPE} = 100 \times \frac{\sum_{j=1}^{T}\sum_{i=1}^{S} \mid \tilde{x}_{s_i,t_j} - x_{s_i,t_j} \mid}{\sum_{j=1}^{T}\sum_{i=1}^{S} \mid x_{s_i,t_j} \mid}, \tag{12}$$

where (as it was defined in Section 3.1) $x_{s_i,t_j}\ :\ j = 1,\dots,T;$ $i = 1,\dots,S$ is a spatio-temporal sample from the real series, $\tilde{x}_{s_i,t_j}$ represents the predicted series, S is the total number of traffic sensors and T the total number of predicted timesteps.

For all these metrics, the closer to zero they are the better the performance is. While RMSE already provides a dispersion measure respect to real series, bias is better to find particular predispositions when making predictions. WMAPE is scale independent and can handle 0s in the series, which makes it interesting for comparing different zones.

5. Results

5.1. Error estimation

A general comparison of the different error metrics can be seen in Table 3. Bias is represented by its absolute value. These values correspond to averaging each metric for all spatial zones. Highlighted in bold, CRANN results show a better performance overall for all errors. Even although run time shows to be worse, we do not reckon this issue should be of special practical importance as the traffic spatio-temporal distribution tends to be stable in time. Thus, no frequent retraining is usually needed in an operational setting.

For a better understanding of how each model is performing, Fig. 8 present RMSE and WMAPE error metrics but with their distribution for each zone separately. While LSTM and recurrent models in general are a standard for time series forecasting, our experiments demonstrate that standard CNN can perform similar (or even better) than recurrent models and should have a bigger space in time series. Also, vanilla LSTM might not be the best option for a real world spatio-temporal system with high complexity. Oddly, the CNN+LSTM model performs worse

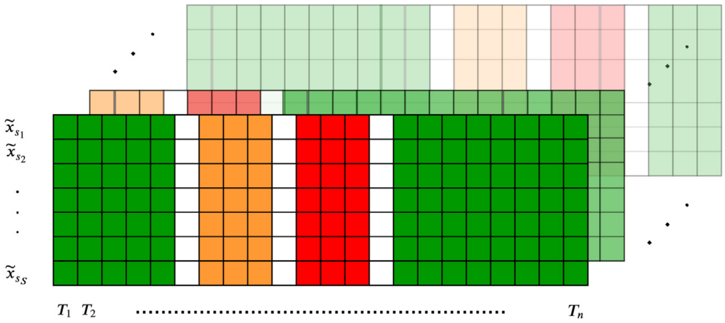


Fig. 7. Validation methodology example with training (green), validation (orange) and, test (red) sets for our proposed cross-validation procedure. Rows shown in white are omitted due to dependency considerations.

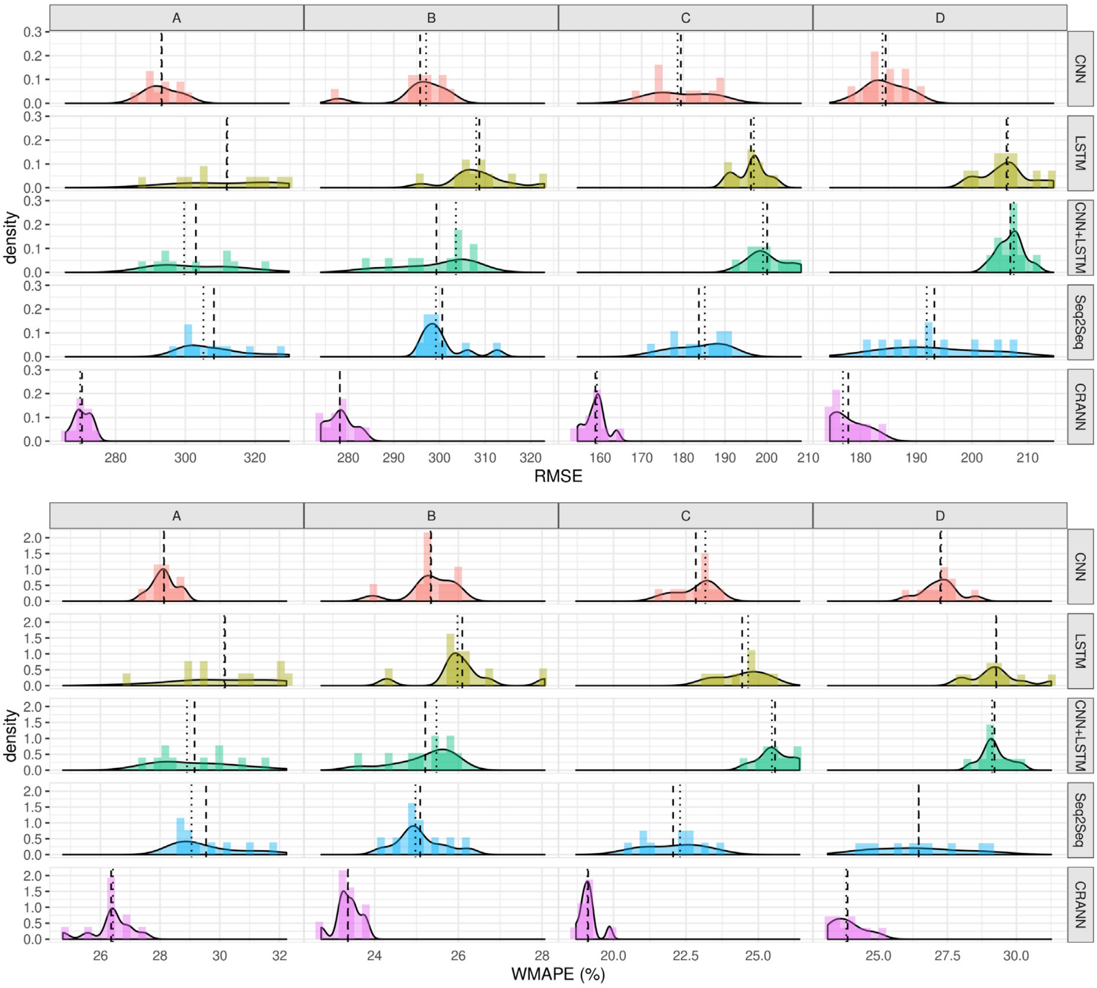


Fig. 8. Distribution of RMSE and WMAPE metrics for each zone and model. Dashed vertical line represents the mean, dotted vertical line represents median.

than the traditional CNN model, which can be due to the LSTM module negatively affecting its behavior. With p-values < 0.05 when comparing with all the baseline models, CRANN can be considered as statistically significantly better at all error metrics with a confidence of 95%.

From Fig. 8 we can also deduce that the deviation of the CRANN model is generally stable and is the smallest one. In fact, models that are highly dependent on a recurrent neural network show a higher-deviation tendency respect to strongly CNN-based models.

Bias exhibits a clear under zero tendency, meaning that all models tend to underestimate their predictions. For a deeper understanding of this phenomenon, Fig. 9 shows CRANN's bias spatial distribution for each studied sensor. Compared with Fig. 6,

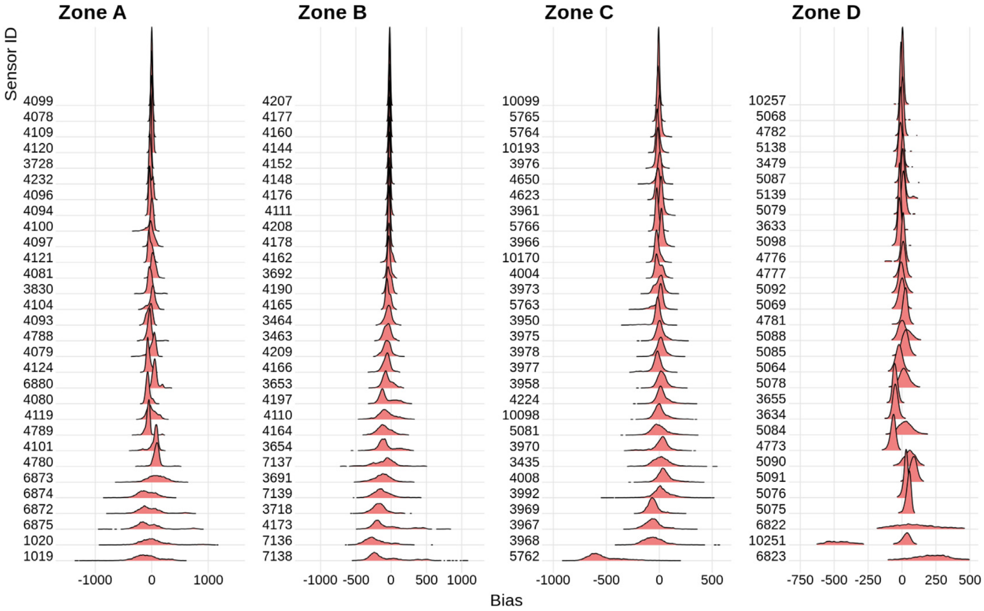


Fig. 9. Bias distribution for each traffic sensor. When compared with Fig. 6 it should be clear that measurement points with higher traffic intensities and more variability are shifted to the left in bias, resulting in an underestimation of the real series.

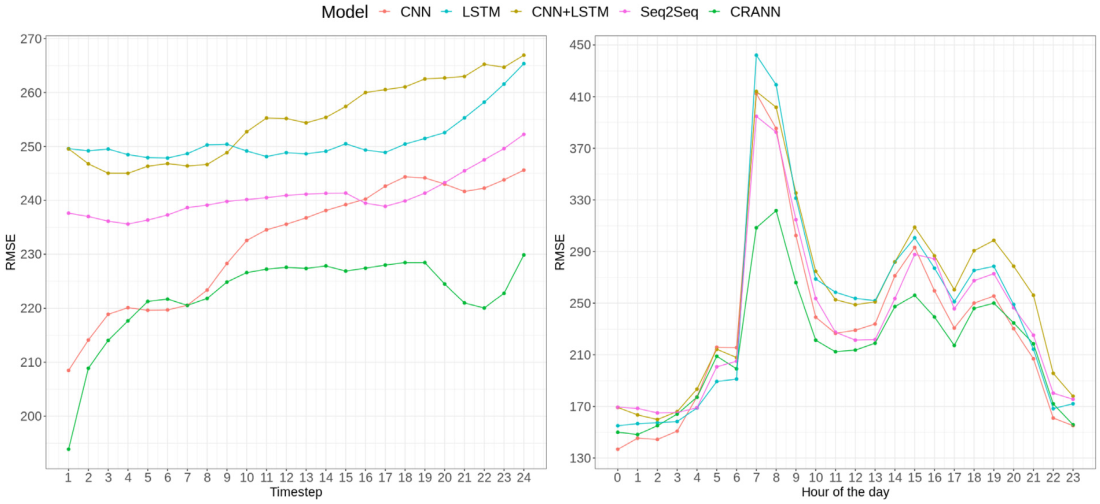


Fig. 10. RMSE analysis for time dimension. How it varies depending on the prediction timestep for all models (left). Average error depending on the hour of the day (right).

it is clear that traffic sensors with higher traffic intensity values, which in turn coincide with those sensors with distributions with greater dispersion, are mainly responsible of this behavior. While we would expect higher errors in these kinds of sensors with such an aggressive traffic pattern, it is not clear why the shifting occurs in only one direction. Nevertheless, as this anomaly happens for all CRANN and baseline models in every zone, we expect that its nature is intrinsic for the system or the validation methodology.

Concerning temporal dimension, a simple analysis can show some expected behavior. As shown in Fig. 10 (left), all models experiment an increase of average RMSE when the predicted timestep goes further, as we could expect. As spot-forecasting is based on evaluation through all possible series, these timesteps do not have a direct correspondence with specific hours of the day and this figure is not contaminated by natural dynamics of traffic. However, there are two clear patterns: LSTM-based models (LSTM, CNN+LSTM, and seq2seq) share a higher error for the first horizons, which are usually considered easier to predict under the hypothesis of inertia of the series. This tells us that they are not capturing this inertia correctly. At the same time, CNN-based models (CNN and CRANN) manage to capture the inertia of the series. Having introduced autoregressive terms into the CRANN model stands as a positive alternative to alleviate and improve this difficulty. Also, we can see a valley from timestep $t + 20$ to $t + 24$ as due to traffic periodicity, that fraction of the series is highly similar to the one introduced as autoregressive terms (timesteps $t - 1$ to $t - 4$).

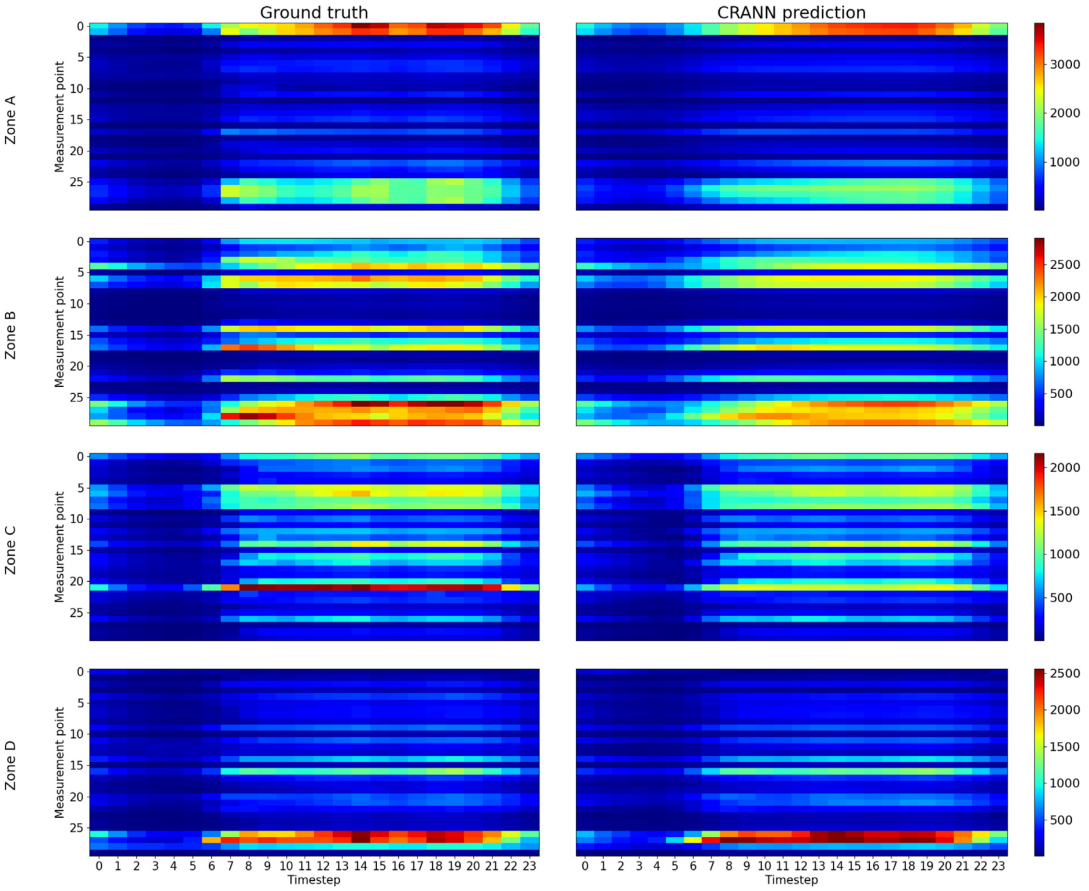


Fig. 11. Example of CRANN's predictions for all zones. Average results for predictions starting at 00:00 and ending at 23:00.

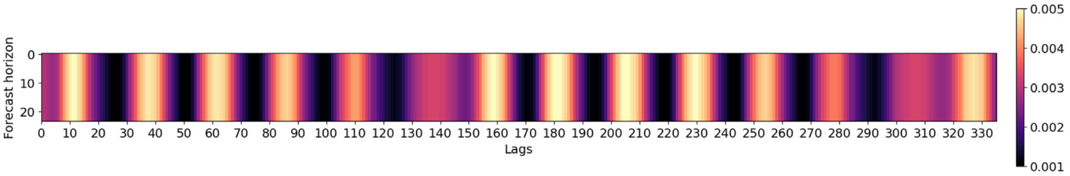


Fig. 12. Average temporal attention given by the temporal module of CRANN. Attention weights are represented as a function of input and output series. In the x-axis, past lags from the input series. In the y-axis, forecast horizons (i.e. future lags) from the output series. A (x, y) value represents how important is timestep x to predict timestep y.

Meanwhile, Fig. 10 (right) let us understand how the average error of the different models are distributed as a function of the hour of the day in which the prediction is being made. As we would expect, these errors are bigger at rush hours, giving us a distribution with same shape than the one presented in Fig. 5. Nevertheless, CRANN model stands out for its ability to outperform significantly its rivals in those exact instants, when it is precisely more useful and challenging to get a good behavior.

Lastly, Fig. 11 displays an example forecast of CRANN. By taking all series starting at 00:00 and ending at 23:00 it is possible to visualize average performance of the model in a specific context. This figure clearly shows how our model is successful in learning the spatio-temporal dynamic of traffic, even adapting its behavior to fine details in a highly complex spatio-temporal problem.

5.2. Interpretability

In order to better understand how our model works, we might use all the interpretability layers presented in Section 3.2.4. Thus, we will first analyze attention in the temporal dimension, then attention in the spatial dimension, and finally the variable importance for all involved features. Also, interpretability will let us corroborate our initial hypothesis about how each module tackles different aspects of spatio-temporal series: trend, seasonality, inertia, and spatial relations.

Starting by our temporal module (see Section 3.2.2), Fig. 12 shows average attention weights computed by the attention mechanism as a function of both input and output timesteps. From this figure, we can have a clear intuition about the 24 h pattern that our model has learned. At the same time, time-back steps 160 and 325, which correspond to 7 and 14 days before

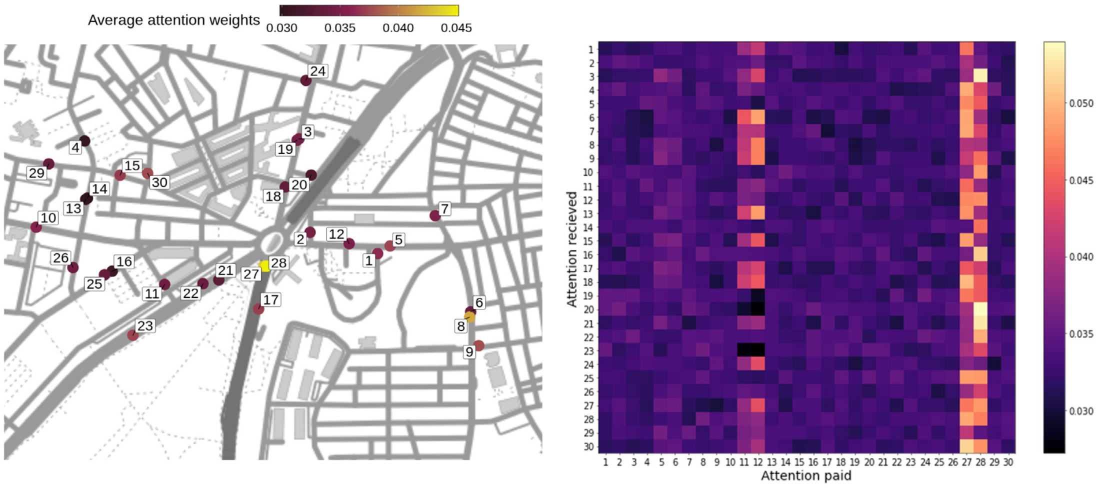


Fig. 13. Average spatial attention given by CRANN spatial module at zone D. Attention weights are averaged for all sensors and timesteps (left). Attention weights for each pair of sensors (right).

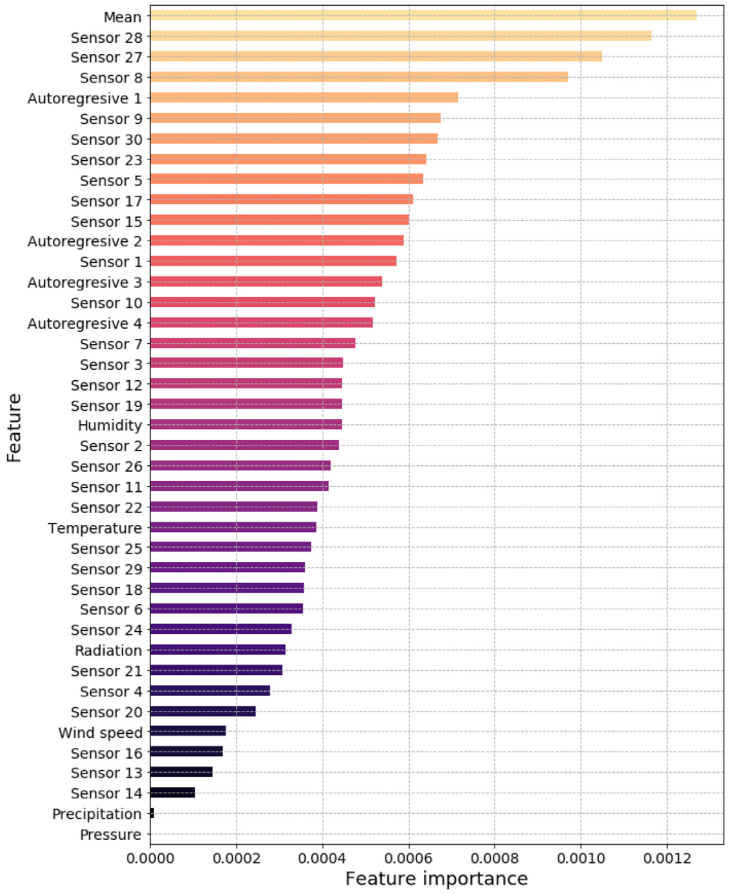


Fig. 14. Mean SHAP values for all features in dense module computed in zone D. Temporal module output (*Mean*), spatial module output (*Sensors*), autoregressive terms, and exogenous variables (both represented by their names).

the prediction, show to be more important as traffic presents a seven days seasonality too. As the input series approach to the forecasting window, the importance keeps growing proving that the temporal module is regulating trend as we were looking for. The fact that no shifting is happening is due to averaging over all test samples.

With respect to the spatial module attention (see Section 3.2.3), it is obviously highly dependent on each specific zone. For that reason, Fig. 13 illustrates the attention weights for traffic sensors in Zone D. As our defined spatial attention mechanism uses different weights for each lag of the input, average values are shown. As it can be seen (left), sensors with especially complex conditions (high traffic intensity, big avenues...) are usually scored as more important by the spatio-temporal attention mechanism. This is the case of points 8, 27, and 28 for example. On the contrary, those points that we would expect to have less impact in global traffic show smaller values, like sensors 4, 16, and 20. Similarly (right), sensors in heavy traffic intensity emplacements show to receive higher attention. As we tackle the long-term forecasting problem, we do not expect our model to pay attention by closeness, but by general importance in the entire zone.

Finally, from average SHAP values computed for the dense module at Zone D (see Section 3.2.4), shown in Fig. 14, we can extract several conclusions. First of all, it supports the idea that using average traffic intensity (''Mean'') for trend and seasonality modeling (temporal module) might be beneficial. Secondly, the importance given to traffic sensors follows a similar pattern to the one seen previously by the spatio-temporal attention mechanism, reinforcing the idea of which spatial points are more important. Thirdly, the autoregressive term that tries to capture the inertia of the series seems to contribute positively too. Lastly, exogenous data importance points out that it has the ability to improve the prediction significantly and should be chosen carefully for each problem.

6. Conclusions and future directions

Through this paper, a new spatio-temporal framework based on attention mechanisms whose operation rest on several spatio-temporal series components is presented. Unlike previous methodologies, we focus our efforts on creating a system that can be considered robust and adaptable, evaluating it in a non-fixed scenario. After being applied to a real traffic dataset, it has been proved that outperforms four state of the art neural architectures and it has been studied its behavior respect to both time and spatial dimension through extensive experimentation. By analyzing four different locations with 30 traffic sensors each, we can confirm the statistical significance of our results with a confidence of 95% for forecasting horizons of up to 24 h.

Thanks to the interpretable nature of the model, we have illustrated how that information might be used in order to understand better how the framework works, how it can give us specific information from the problem domain and why our network architecture is well founded. Concretely, the conducted experiments have shown that, as we postulated, the temporal module regulates seasonality and trend, spatial module is capable of extracting short-term and spatial relations, and that it is necessary to introduce explicit autoregressive terms to exploit inertia correctly. Finally, these experiments demonstrate the effectiveness of all these terms to make the final prediction.

For future work, it might be interesting to evaluate the proposed method over a wider range of series in order to generalize the results and see its behavior over different applications. With the actual ability from the spatial module to model attention for both input dimensions, space and time, it could be beneficial to extend these idea to outputs dimensions too, having different attention weights for different predicted timesteps. Lastly, it should be studied how to use exogenous spatio-dependent data in the best possible way.

CRediT authorship contribution statement

Rodrigo de Medrano: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - review & editing, Visualization. **José L. Aznarte:** Conceptualization, Methodology, Validation, Formal analysis, Investigation, Resources, Data curation, Writing - review & editing, Visualization, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research has been partially funded by the *Empresa Municipal de Transportes* (EMT) of Madrid, Spain under the program *"Aula Universitaria EMT/UNED de Calidad del Aire y Movilidad Sostenible"*.

References

[1] Digital Earth, Library Catalog: www.digitalearth.art URL https://www.digitalearth.art.

[2] Galileo is the European global satellite-based navigation system | European Global Navigation Satellite Systems Agency URL https://www.gsa.europa.eu/european-gnss/galileo/galileo-european-global-satellite-based-navigation-system.

[3] Y. Wang, M. Long, J. Wang, Z. Gao, P.S. Yu, Predrnn: Recurrent neural networks for predictive learning using spatiotemporal LSTMs, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 879–888.

[4] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, S. Savarese, Social LSTM: Human trajectory prediction in crowded spaces, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 961–971, http://dx.doi.org/10.1109/CVPR.2016.110, ISSN: 1063-6919.

[5] J. Ke, H. Yang, H. Zheng, X. Chen, Y. Jia, P. Gong, J. Ye, Hexagon-based convolutional neural network for supply-demand forecasting of ride-sourcing services, IEEE Trans. Intell. Transp. Syst. (ISSN: 1558-0016) 20 (11) (2019) 4160–4173, http://dx.doi.org/10.1109/TITS.2018.2882861, Conference Name: IEEE Transactions on Intelligent Transportation Systems.

[6] Q. Zhu, J. Chen, L. Zhu, X. Duan, Y. Liu, Wind speed prediction with spatio–temporal correlation: A deep learning approach, Energies 11 (4) (2018) 705, http://dx.doi.org/10.3390/en11040705, Number: 4 Publisher: Multidisciplinary Digital Publishing Institute.

[7] B. Liao, F. Wu, J. Zhang, M. Cai, S. Tang, Y. Gao, C. Wu, S. Yang, W. Zhu, Y. Guo, Dest-resnet: A deep spatiotemporal residual network for hotspot traffic speed prediction, in: 2018 ACM Multimedia Conference on Multimedia Conference - MM '18, ACM Press, Seoul, Republic of Korea, ISBN: 978-1-4503-5665-7, 2018, pp. 1883–1891, http://dx.doi.org/10.1145/3240508.3240656.

[8] H.-F. Yang, T.S. Dillon, Y.-P.P. Chen, Optimized structure of the traffic flow forecasting model with a deep learning approach, IEEE Trans. Neural Netw. Learn. Syst. (ISSN: 2162-2388) 28 (10) (2017) 2371–2381, http://dx.doi.org/10.1109/TNNLS.2016.2574840, Conference Name: IEEE Transactions on Neural Networks and Learning Systems.

[9] M.-T. Luong, H. Pham, C.D. Manning, Effective approaches to attention-based neural machine translation, 2015, arXiv:1508.04025 [cs] ArXiv:1508.04025.

[10] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, 2016, arXiv:1409.0473 [cs, stat] ArXiv:1409.0473.

[11] W. Cheng, Y. Shen, Y. Zhu, L. Huang, A neural attention model for urban air quality inference: Learning the weights of monitoring stations, in: S.A. McIlraith, K.Q. Weinberger (Eds.), Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th Innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, AAAI Press, 2018, pp. 2151–2158.

[12] Y. Chen, G. Peng, Z. Zhu, S. Li, A novel deep learning method based on attention mechanism for bearing remaining useful life prediction, Appl. Soft Comput. (ISSN: 1568-4946) 86 (2020) 105919, http://dx.doi.org/10.1016/j.asoc.2019.105919.

[13] X. Fu, F. Gao, J. Wu, X. Wei, F. Duan, Spatiotemporal attention networks for wind power forecasting, 2019, arXiv:1909.07369 [cs] ArXiv:1909.07369.

[14] S. Wang, J. Cao, P.S. Yu, Deep learning for spatio-temporal data mining: A survey, 2019, arXiv:1906.04928 [cs, stat] ArXiv:1906.04928.

[15] M. Hamed Mohammad, R. Al-Masaeid Hashem, M.B. Said Zahi, Short-term prediction of traffic volume in urban arterials, J. Transp. Eng. 121 (3) (1995) 249–254, http://dx.doi.org/10.1061/(ASCE)0733-947X(1995)121:3(249), Publisher: American Society of Civil Engineers.

[16] S.V. Kumar, L. Vanajakshi, Short-term traffic flow prediction using seasonal ARIMA model with limited input data, Eur. Trans. Res. Rev. (ISSN: 1866-8887) 7 (3) (2015) 21, http://dx.doi.org/10.1007/s12544-015-0170-8.

[17] C.M. Queen, C.J. Albers, Intervention and causality: Forecasting traffic flows using a dynamic Bayesian network, J. Amer. Statist. Assoc. (ISSN: 0162-1459) 104 (486) (2009) 669–681, http://dx.doi.org/10.1198/jasa.2009.0042.

[18] A. Pascale, M. Nicoli, Adaptative Bayesian network for traffic flow prediction, 2011.

[19] X. Dong, T. Lei, S. Jin, Z. Hou, Short-term traffic flow prediction based on xgboost, in: 2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS), IEEE, Enshi, ISBN: 978-1-5386-2618-4, 2018, pp. 854–859, http://dx.doi.org/10.1109/DDCLS.2018.8516114.

[20] W. Alajali, W. Zhou, S. Wen, Traffic flow prediction for road intersection safety, in: 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), IEEE, Guangzhou, China, ISBN: 978-1-5386-9380-3, 2018, pp. 812–820, http://dx.doi.org/10.1109/SmartWorld.2018.00151.

[21] T. Wu, K. Xie, G. Song, C. Hu, A multiple SVR approach with time lags for traffic flow prediction, in: 2008 11th International IEEE Conference on Intelligent Transportation Systems, 2008, pp. 228–233, http://dx.doi.org/10.1109/ITSC.2008.4732663, ISSN: 2153-0009, 2153-0017.

[22] Z. Mingheng, Z. Yaobao, H. Ganglong, C. Gang, Accurate multisteps traffic flow prediction based on SVM, 2013, http://dx.doi.org/10.1155/2013/418303.

[23] M.C. Ho, J.M.-Y. Lim, K.L. Soon, C.Y. Chong, An improved pheromone-based vehicle rerouting system to reduce traffic congestion, Appl. Soft Comput. (ISSN: 1568-4946) 84 (2019) 105702, http://dx.doi.org/10.1016/j.asoc.2019.105702.

[24] Z. Liu, Z. Li, K. Wu, M. Li, Urban traffic prediction from mobility data using deep learning, IEEE Netw. (ISSN: 0890-8044) 32 (4) (2018) 40–46, http://dx.doi.org/10.1109/MNET.2018.1700411.

[25] A. Ermagun, D. Levinson, Spatiotemporal traffic forecasting: review and proposed directions, Trans. Rev. 38 (6) (2018) 786–814, http://dx.doi.org/10.1080/01441647.2018.1442887, ISSN: 0144-1647, 1464-5327.

[26] Y. Wu, H. Tan, L. Qin, B. Ran, Z. Jiang, A hybrid deep learning based traffic flow prediction method and its understanding, Trans. Res. Part C (ISSN: 0968-090X) 90 (2018) 166–180, http://dx.doi.org/10.1016/j.trc.2018.03.001.

[27] L.N.N. Do, H.L. Vu, B.Q. Vo, Z. Liu, D. Phung, An effective spatial-temporal attention based neural network for traffic flow prediction, Trans. Res. Part C (ISSN: 0968-090X) 108 (2019) 12–28, http://dx.doi.org/10.1016/j.trc.2019.09.008.

[28] S. Guo, Y. Lin, S. Li, Z. Chen, H. Wan, Deep spatial–temporal 3D convolutional neural networks for traffic data forecasting, IEEE Trans. Intell. Transp. Syst. 20 (10) (2019) 3913–3926, http://dx.doi.org/10.1109/TITS.2019.2906365.

[29] T. Bogaerts, A.D. Masegosa, J.S. Angarita-Zapata, E. Onieva, P. Hellinckx, A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data, Trans. Res. Part C (ISSN: 0968-090X) 112 (2020) 62–77, http://dx.doi.org/10.1016/j.trc.2020.01.010.

[30] S. Deng, S. Jia, J. Chen, Exploring spatial–temporal relations via deep convolutional neural networks for traffic flow prediction with incomplete data, Appl. Soft Comput. (ISSN: 1568-4946) 78 (2019) 712–721, http://dx.doi.org/10.1016/j.asoc.2018.09.040.

[31] L. Qu, W. Li, W. Li, D. Ma, Y. Wang, Daily long-term traffic flow forecasting based on a deep neural network, Expert Syst. Appl. (ISSN: 0957-4174) 121 (2019) 304–312, http://dx.doi.org/10.1016/j.eswa.2018.12.031.

[32] Z. He, C.-Y. Chow, J.-D. Zhang, STCNN: A spatio-temporal convolutional neural network for long-term traffic prediction, in: 2019 20th IEEE International Conference on Mobile Data Management (MDM), 2019, pp. 226–233, http://dx.doi.org/10.1109/MDM.2019.00-53, ISSN: 1551-6245.

[33] C. Bergmeir, R. Hyndman, J. Benítez, Bagging exponential smoothing methods using STL decomposition and box–cox transformation, Int. J. Forecast. (ISSN: 0169-2070) 32 (2) (2016) 303–312, http://dx.doi.org/10.1016/j.ijforecast.2015.07.002.

[34] E.M. de Oliveira, F.L. Cyrino Oliveira, Forecasting mid-long term electric energy consumption through bagging ARIMA and exponential smoothing methods, Energy (ISSN: 0360-5442) 144 (2018) 776–788, http://dx.doi.org/10.1016/j.energy.2017.12.049.

[35] Z. Li, Y. Li, L. Li, A comparison of detrending models and multi-regime models for traffic flow prediction, IEEE Intell. Trans. Syst. Mag. 6 (4) (2014) 34–44.

[36] X. Dai, R. Fu, E. Zhao, Z. Zhang, Y. Lin, F.-Y. Wang, L. Li, Deeptrend 2.0: A light-weighted multi-scale traffic prediction model using detrending, Trans. Res. Part C (ISSN: 0968-090X) 103 (2019) 142–157, http://dx.doi.org/10.1016/j.trc.2019.03.022.

[37] G. Lai, W.-C. Chang, Y. Yang, H. Liu, Modeling long- and short-term temporal patterns with deep neural networks, 2018, arXiv:1703.07015 [cs] ArXiv:1703.07015.

[38] B. Liu, X. Tang, J. Cheng, P. Shi, Traffic Flow Combination Forecasting Method Based on Improved LSTM and ARIMA Int. J. Embedded Syst. 12 (1) 22–30.

[39] K. Bandara, C. Bergmeir, H. Hewamalage, LSTM-msnet: Leveraging forecasts on sets of related time series with multiple seasonal patterns, 2019, arXiv:1909.04293 [cs, stat] ArXiv:1909.04293.

[40] M. Nelson, T. Hill, W. Remus, M. O'Connor, Time series forecasting using neural networks: should the data be deseasonalized first? J. Forecast. (ISSN: 1099-131X) 18 (5) (1999) 359–367, http://dx.doi.org/10.1002/(SICI)1099-131X(199909)18:5<359::AID-FOR746>3.0.CO;2-P.

[41] R. Asadi, A.C. Regan, A spatio-temporal decomposition based deep neural network for time series forecasting, Appl. Soft Comput. (ISSN: 1568-4946) 87 (2020) 105963, http://dx.doi.org/10.1016/j.asoc.2019.105963.

[42] O. Li, H. Liu, C. Chen, C. Rudin, Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions 8.

[43] Q. You, H. Jin, Z. Wang, C. Fang, J. Luo, Image captioning with semantic attention, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[44] J. Wang, R. Chen, Z. He, Traffic speed prediction for urban transportation network: A path based deep learning approach, Trans. Res. Part C (ISSN: 0968-090X) 100 (2019) 372–385, http://dx.doi.org/10.1016/j.trc.2019.02.002.

[45] Z. Cui, K. Henrickson, R. Ke, X. Dong, Y. Wang, High-order graph convolutional recurrent neural network: a deep learning framework for network-scale traffic learning and forecasting, 2019, Number: 19-05236.

[46] K. Cho, B. van Merriënboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: Encoder–decoder approaches, in: Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Association for Computational Linguistics, Doha, Qatar, 2014, pp. 103–111, http://dx.doi.org/10.3115/v1/W14-4012.

[47] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org, 2017, pp. 3319–3328.

[48] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Advances in Neural Information Processing Systems, 2017, pp. 4765–4774.

[49] C. Bergmeir, R.J. Hyndman, B. Koo, A note on the validity of cross-validation for evaluating autoregressive time series prediction, Comput. Statist. Data Anal. (ISSN: 01679473) 120 (2018) 70–83, http://dx.doi.org/10.1016/j.csda.2017.11.003.

Chapter 4

On the inclusion of spatial information for spatio-temporal neural networks

Type:	Published article
Journal:	Neural Computing & Applications
Authors:	Rodrigo de Medrano & José Luis Aznarte
Published:	May 2021
Impact factor:	4.774
5-Year Impact factor:	4.627
Quartile:	Q1 (Artificial Intelligence)
DOI:	10.1007/s00521-021-06111-6
Contribution:	Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing.

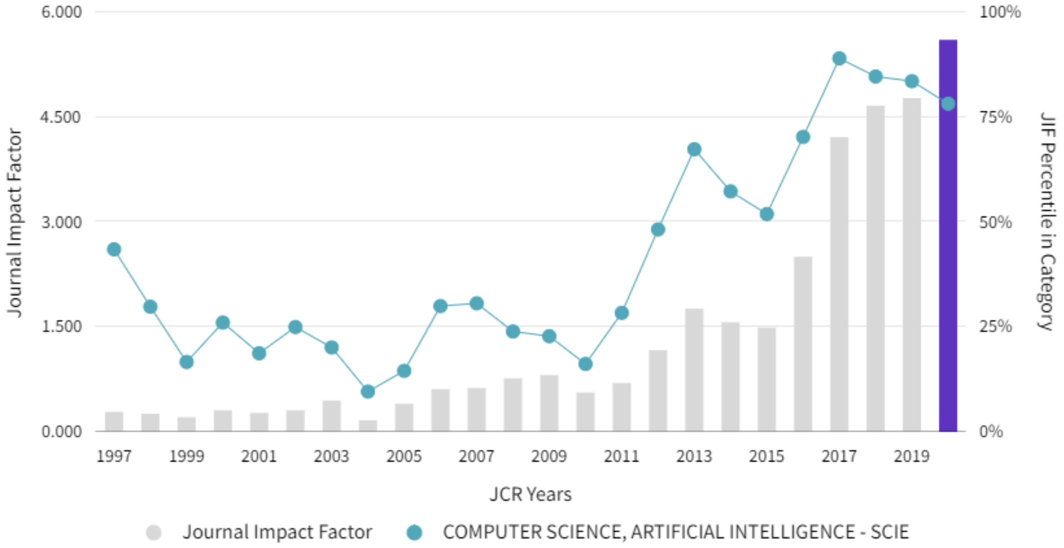


FIGURE 4.1: Impact factor: Neural Computing & Applications.

Neural Computing and Applications
https://doi.org/10.1007/s00521-021-06111-6

ORIGINAL ARTICLE

On the inclusion of spatial information for spatio-temporal neural networks

Rodrigo de Medrano[1] · José L. Aznarte[1]

Received: 29 October 2020 / Accepted: 5 May 2021

Abstract

When confronting a spatio-temporal regression, it is sensible to feed the model with any available *prior* information about the spatial dimension. For example, it is common to define the architecture of neural networks based on spatial closeness, adjacency, or correlation. A common alternative, if spatial information is not available or is too costly to introduce it in the model, is to learn it as an extra step of the model. While the use of *prior* spatial knowledge, given or learned, might be beneficial, in this work we question this principle by comparing traditional forms of convolution-based neural networks for regression with their respective spatial agnostic versions. Our results show that the typical inclusion of *prior* spatial information is not really needed in most cases. In order to validate this counterintuitive result, we perform thorough experiments over ten different datasets related to sustainable mobility and air quality, substantiating our conclusions on real world problems with direct implications for public health and economy. By comparing the performance over these datasets between traditional and their respective agnostic models, we can confirm the statistical significance of our findings with a confidence of 95%.

Keywords Neural networks · Spatio-temporal series · Spatial dimension · Convolutional neural networks · Regression

1 Introduction

Convolutional neural networks (CNN) are well known for their ability to handle spatial data in several contexts, like images or spatial phenomena. However, in the last few years, they have demonstrated to hold a good position also when dealing with temporal data. Thus, they are widely used in spatio-temporal regression problems, with outstanding behavior when coping with both spatial and temporal dimensions.

Due to its parameter structure, CNNs are usually employed when it is possible to order input data in a grid. Furthermore, they treat each location equally, learning and sharing the same weights for all spatial points. Given that it is not rare that the phenomenon under study presents the same nature all over the grid, in a wide range of applications this property is a clear advantage in order to minimize the number of parameters and calculations for learning a specific task. This leads to good performance with fewer resources compared to feedforward neural networks (FNN) and recurrent neural networks (RNN). For example, pollution and traffic regression share an approximately equivalent temporal behavior and distribution at each location (at least in a close environment), meaning that it is possible to share parameters and get a smooth approximation for these phenomena via traditional CNNs.

However, this property of CNNs (which is usually known as *equivariance*), might not always be the best deal when solving some typical problems: sometimes, although similar, treating all locations equally does not hold as a valid or acceptable hypothesis and so, learning a spatial shared-based representation might not be the best option if the system representation is not chosen carefully. In the previous example, it is obvious that different traffic sensors or pollution stations will have different properties, even

✉ José L. Aznarte
jlaznarte@dia.uned.es

Rodrigo de Medrano
rdemedrano@dia.uned.es

[1] Artificial Intelligence Department, Universidad Nacional de Educación a Distancia—UNED, 28041 Madrid, Spain

Published online: 19 May 2021

though their temporal dynamic will be somehow similar. For spatio-temporal regression specifically, several proposals have been made in order to tackle this problem, but two of them stand out for their wide acceptance:

- Order the grid by Euclidean distance (from now on, just closeness) and use CNNs.
- Define the system in a graph structure and model it via graph convolutional networks (GCN).

In both cases, a common assumption is made: closer locations have similar properties and, because of that, the shared-weights learned by the networks are more reliable. This way, the spatial dimension in CNNs keeps a low number of parameters. In other words, the idea is to induce a spatial bias that will help the network to learn and make predictions. These types of strategies, based on providing information about the problem externally, are very common but generates a dependence on the accuracy or validity of this information to model the reality of the problem.

However, these solutions do have some disadvantages. First, they do not completely solve the fact that each location, although related to the rest, has its own properties. Even more, although the assumption that closer locations behave similarly is usually blindly accepted, this might not hold always for real problems: not only depends on the phenomenon, but also on the temporal and spatial granularity with which the data are taken. For example, [1] shows how the small number of pollution stations that monitor most cities implies a spatially sparse representation that might not be enough to obtain a reliable spatiotemporal distribution based on station's adjacency of air pollutant concentrations. Thus, the benefits of learning a latent representation based on sharing parameters are conditioned by the particularities of each specific problem and, contrary to popular belief, the spatial proximity between locations is not necessarily the main factor. Second, in both cases, it is necessary to introduce *prior* spatial knowledge to the system, making it less 'intelligent' and more laborious to work with.

In this paper, we focus on whether defining adjacency-based convolutional architectures for regression problems is as important or positive as has traditionally been assumed. To explore and contrast our hypothesis, we propose to compare a set of widely used traditional convolutional methods with their respective spatial agnostic versions. Here, the denotation "spatial agnostic" makes reference at not including specific mechanisms that exploit spatial information explicitly. By showing that no improvement is reported when using *prior* spatial knowledge, we can reject the idea that models with a spatial bias will result systematically in better forecasters. Also, models with spatial agnostic nature can be a suitable choice when spatial information is not easily achievable or within reach.

What happens if we closely examine the temporal dimension? In multiple real applications in which this spatial agnosticism does not exactly hold, temporal equivalence between locations is more plausible: temporal distributions along spatial points might better fulfill the assumption of sharing parameters compared to the spatial dimension. This means that, while it is common to use some sort of recurrent module to model temporal relations, convolutions can be perfectly valid candidates for this work, using a lower number of parameters. Thus, sharing parameters for all locations between subsequent past timesteps in the temporal dimension might work better than in the spatial dimension.

To validate our hypothesis, we compare several models and dig deeper into the real importance of closeness relations through extensive experimentation. For this purpose, the vast field of air quality and sustainable mobility has been chosen. With a wide number of long spatio-temporal series with spatial particularities but approximately equivalent temporal dynamics (due to their relation with human behavior) and high nonlinearity, it is a perfect field to corroborate our hypotheses. Since it is considered of great importance for public health and also to economy, it is potentially beneficial to have simpler and easily deployable models in this field. Also, our work is a good starting point to rethink the way of working with spatio-temporal series if we want to extract and make use of the spatial information of the problem in a more efficient and simple way beyond classical adjacency hypothesis. Thus, it would be plausible to improve not just performance but to gain insight into real systems when working with spatio-temporal neural networks.

The main contributions of this study are summarized as follows:

- We delve into the counterintuitive idea that including spatial relations based on closeness are not necessarily the optimal option when working with neural networks for regression in spatio-temporal problems. Concretely, we compare several traditional methodologies with their respective spatial agnostic version.
- The contribution is illustrated by tackling a variety of prediction problems related to air quality and sustainable mobility. All of them are considered of great importance and significantly complex for both spatial and temporal dimensions.
- Results show that spatial agnostic methods equal state-of-the-art models in accuracy without the need of *prior* spatial information.

The rest of the paper is organized as follows: related work is discussed in Sect. 2, while Sect. 3 presents the methods and all needed theory for this work. Then, in Sect. 4 we introduce our datasets, experimental design, and their properties. Section 5 illustrates the evaluation of the neural architectures as derived after appropriate experimentation. Finally, in Sect. 6 we point out conclusions from our work.

2 Related work

2.1 The rise of convolutions

Since CNNs were proposed as neural architectures [2], they have shown to handle especially well spatially-ordered data. During the last decades, these kind of neural networks have grown in importance, becoming one of the most used neural paradigms for a wide number of applications.

In the case of intrinsic 2D problems, like images, CNNs have turned out to be the option per excellence. Concretely, with [3] started a reign of CNN for computer vision problems. Not much later, the idea that weight sharing could lead to potentially suboptimal performance for some images, like portraits, was studied [4]. In the present, CNNs are widely used for this kind of problem and have been well characterized.

However, CNNs are not constrained to natural 2D systems. For example, time series seen as a 1D sequence have been handled by convolutional models with good results [5, 6]. Spatio-temporal series have growth in importance, and CNNs have been well studied and are already a standard when dealing with this kind of series [7, 8]. A similar field to spatio-temporal series is video-sequence analysis, where both spatial and temporal relations need to be modeled [9]. Within this last topic, some examples in which parameter sharing are indeed highly positive can be found, as for example enhancing video spatial resolution for creating smooth results [10] or action recognition [11].

2.2 Spatial dimension in spatio-temporal neural networks

In spatio-temporal regression specifically, convolutional-based networks are one of the leading options too. As explained in Sect. 1, convolution shines in a wide range of applications involving physical spatial locations. However, how this dimension is treated by the convolution has not received particular attention. Thus, we have several options that are widely used but not necessarily optimal.

For example, in traffic forecasting, defining your space as a natural grid. [12] is a good example of 2D image-to-image prediction problem in which, by using channels as timesteps and 3D kernels, spatio-temporal relations are exploited. As average traffic speeds for each road segment is used, no need for *prior* spatial information is needed, and the grid arrangement is natural. However, closer areas are not necessarily more related. In [13], it is shown that the 3D convolution might work better, but the same spatial arrangements and assumptions are made. Another example, but in the bike-sharing regression problem, would be [14]. In this case, the claim that the spatiotemporal distribution is endogenously dependent on the zonal attribute of adjacent areas makes sense as they use a grid big enough of 4×4 km. However, if the granularity of the spatial dimension is thinner, adjacent bike stations are not expected to be especially related, as very short trips (barely hundreds of meters, which is what usually distances stations) are not common, and by so, their argument does not necessarily hold.

When measurement points are directly used as an arrangement for the spatial dimension, not only it is necessary to impose the same closeness supposition as before, but a special treatment is usually needed to arrange locations correctly. Some examples are [15], where the authors order traffic sensors in a 1D grid; or [16], where measurement points are ordered as 2D images.

In recent years, graphs-based networks have received increasing attention. GCNs not only have shown a very competitive performance, but a graph structure is more suitable than grids for some specific problems where relations might be non-Euclidean and directional [17]. Among the different convolutions in graphs, all of them depend heavily on an adjacency matrix which usually needs to be manually defined. This adjacency matrix is of great importance as it defines the graph relations and structure. Depending on the proposal, this matrix might be defined differently: it usually is defined by spatial closeness [18, 19], but there are no restrictions. For example, in [20] temporal trend information is integrated when forming this matrix and, this way, it is expected to take full advantage of both temporal and spatial dimensions. Thus, the adjacency matrix makes the spatial dependency more localized than the plain version. While this freedom to define the adjacency matrix might help to avoid the closeness assumption, it would force you to find which *prior* information may be more optimal for your particular problem. If compared to traditional CNN, GCN presents another advantage: they can naturally process information from a $K-$hop neighborhood [21], not restricting themselves to uniquely adjacent nodes.

Temporal relations with neural networks are usually constrained to using some kind of RNNs. Although many proposals have been done, through this work we will not focus on this broad topic, and we will limit its use to standards.

2.3 Non-locally dependent proposals

The idea that a fixed arrangement for learning spatial relations might not be the best deal is not new in spatio-temporal series forecasting. Lu et al [22] state that "the existence of spatial heterogeneity imposes great influence on modeling the extent and degree of road traffic correlation, which is usually neglected by the traditional distance-based method", and proposed a data-driven approach to measure these correlations. From this starting point, we can select several works that have contributed to refine and depend less on *prior* information in the spatial dimension using neural models.

By using a hierarchical clustering over the spatio-temporal data, [23] refines spatial relations. However, it uses a distance matrix in the process, introducing the aforementioned bias by closeness. In [24], a lasso methodology is used to obtain a sparse model of the system dynamics, which simultaneously identifies spatial correlation along with model parameters.

Attention mechanisms, which appeared on the deep learning scene a few years ago, are a natural way to learn relations beyond the network's original assumptions. In this context, several works have used attention weights to improve performance and demonstrate the correctness of their work with both, grid structure [25] and graph structure [26]. However, [16] shows how closer locations are not necessarily more related, and depending on the problem and the characteristics of the regression, other considerations might be more important when learning spatial relations.

Closer to our work, in [27] a similar issue but with general multivariate time series forecasting is put on the table: existing methods usually fail to fully exploit latent spatial dependencies between pairs of variables and GCNs require well-defined graph structures which means they cannot be applied directly for multivariate time series where the dependencies are not known in advance. In their proposal, they construct a new model that tackles both problems. [28] focus its efforts on dealing with the fact that different spatial locations might have at some degree different dynamics by using traditional CNNs but with the introduction of learnable local inputs/latent variables and learnable local transformations of the inputs.

In the end, all these works focus their attention on solving a specific regression problem, but not delve into how the spatial dimension should be really treated. Furthermore, all these methodologies have in common the need to make their models considerably more complex in order to overcome spatial agnosticism, generally starting from usual convolution operators and refining themselves via extra mechanisms or modules.

3 Methods

Through this section, we present all the theoretical methods and foundations on which our study bases its ideas and experiments on the role of spatial agnosticism in spatio-temporal series. The code for this paper is available in https://github.com/rdemedrano/SANN.

3.1 Preliminaries

As we intend to demonstrate how the typical intrinsic spatial information given to different forms of convolutional methods is not as important as always assumed, we focus this paper on comparing traditional models with their respective agnostic version. Before explaining these methodologies, we introduce some general aspects.

Given a spatio-temporal sequence X, let us call N to the total number of timesteps and S the total number of spatial points. With this notation, a spatio-temporal sample from the series writes as $x_{t_i,s_j} : i = 1,\ldots,T; j = 1,\ldots,S$, being T the total number of timesteps conforming the sample. X_{t} is the slice of series X for timestep t at all locations, and $X_{T,j}$ is the slice of series X in location j for all timesteps. The predicted series is represented by $\tilde{x}_{t'_i,s_j} : i = 1,\ldots, T'; j = 1,\ldots,S$, where T' is the total number of predicted timesteps. We assume that the number of spatial locations is always the same for both the input and output series.

For all models, the input sequence scheme relies upon a $C \times T \times S$ format images as shown in Fig. 1, where rows represent timesteps, columns define themselves spatial locations, and the number of channels C represents the number of input spatio-temporal variables. During this paper, we will work with $C = 1$ (the studied series by

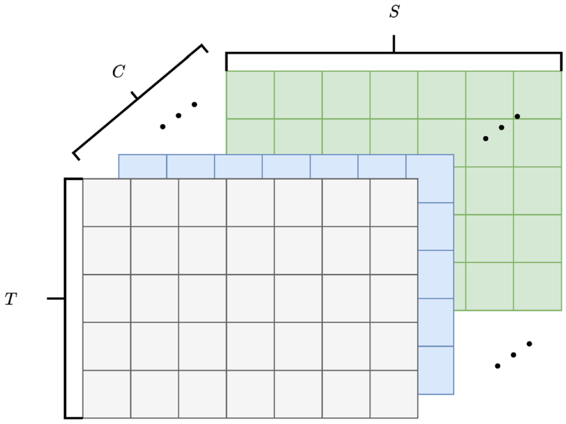


Fig. 1 Input sequence schematic. As long as all variables are spatio-temporal and have an equivalent structure for both dimensions, these sequences can be easily introduced as $C \times T \times S$ images, with variable, temporal, and spatial dimensions, respectively

itself), but is easily extensible to any value. Also, all models will consist in one convolution layer outputting a $T \times S \times H$ tensor, where H is the dimension of the new hidden state or number of new channels. This approach allows us to standardize the input and output format of the convolution layers for all methods.

3.2 Traditional convolutional networks

Now we present the traditional format of convolution-based networks for spatio-temporal series, and we detail how they will be used for testing our main hypothesis.

3.2.1 Convolutional neural networks (CNN)

Convolutional Neural Networks are based on the idea of the convolution operation. Convolution itself ($*$ operator) has the following form for 2D images:

$$(x * K)(i,j) = \sum_{m}^{k_1} \sum_{n}^{k_2} x(m,n)K(i-m,j-n) \tag{1}$$

where K is the kernel. Thus, CNNs are characterized by learning a series of filters which values depend on how adjacent elements are related.

In its classical form, CNNs for spatio-temporal regression rely on ordering the input sequence by spatial adjacency or closeness. Thus, for each row of Fig. 1, spatial zones are mapped into the input tensor in such a way that closer locations are closer in the sequence. By doing so, we make sure that the learnable kernels can take advantage of this *prior* spatial information. The strategies that can be used to adapt convolutional networks to spatio-temporal problems by exploiting this spatial bias based on proximity are multiple (see Sect. 2.2). In our case, the input to the network will be defined as shown in Fig. 2, ordered by spatial location through columns and temporal dimension through rows. Similarly, in the temporal dimension (columns) the kernel gathers adjacent timesteps.

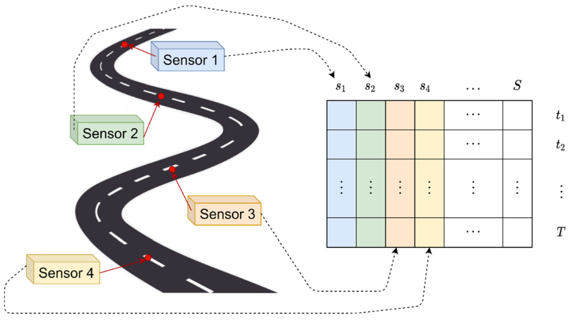


Fig. 2 Example of input tensor definition. Given a network of traffic sensors and its historical series, the objective is to predict future timesteps for all locations using CNNs. The input sequence order in the spatial dimension is usually defined by a logical arrangement of the relative position of the sensors in the network. Traditionally, it is expected to improve network learning through this strategy, generating softer filters by exploiting adjacency relationships

3.2.2 ConvLSTM neural networks

Long Short-Term Memory (LSTM) is a type of recurrent neural network architecture usually used when handling time series data with temporal auto-correlations. An LSTM Neural Network consists of an input gate, an output gate, a memory cell, and a forget gate. During the training phase, a weighted function is learned in each of the gates in order to control how much the network “memorize” and “forget”.

Based on this model, the ConvLSTM model is a variation of LSTM capable of handling spatio-temporal processes [29]. Comparing with the original LSTM model, the input-to-state and state-to-state transitions of the ConvLSTM cell involves convolutional operations, making it a well fit for spatial relations. This model is governed by the following equations.

$$\begin{aligned} i_t &= \sigma(W_{\mathrm{xi}} * X_t + W_{\mathrm{hi}} * H_{t-1} + b_i) \\ f_t &= \sigma(W_{\mathrm{xf}} * X_t + W_{hf} * H_{t-1} + b_f) \\ C_t &= f_t \circ C_{t-1} + tanh(W_{\mathrm{xc}} * X_t + W_{\mathrm{hc}} * H_{t-1} + b_c) \\ o_t &= \sigma(W_{\mathrm{xo}} * X_t + W_{\mathrm{ho}} * H_{t-1} + b_o) \\ H_t &= o_t \circ tanh(C_t) \end{aligned} \tag{2}$$

As previously $*$ denotes the convolution operation while $\circ$ denotes the Hadamard product. Furthermore, for timestep t, we find that i_{t}, f_{t}, o_{t} are the outputs of input gate, forget gate, and output gate, respectively, C_{t} is the cell output, and H_{t} is the hidden state of a cell.

As explained with CNNs (3.2.1), ConvLSTM input sequence is usually ordered by closeness or adjacency in order to take advantage of the shared-weight scheme of the convolution. Through this work, ConvLSTM will use the same input scheme as CNNs as presented in Fig. 2. Thus, convolution operations can make use of this type of spatial relationship.

3.2.3 Graph convolutional neural networks (GCN-LSTM)

While several proposals have been made during the last years to convolute over graphs, we focus on a particular type presented in [21] called High-Order and Adaptive Graph Convolution, as it has shown good performance in a wide variety of problems. In words of the authors, given a graph $\mathcal{G}$, the k-hop (k-th order) neighborhood is defined as: $N_j = \{v_i \in V | d(v_i, v_j) \leq k\}$ for node v_j . In fact, the exact k-hop connectivity can be obtained by the multiplication of the adjacency matrix A, giving as a result A^k. The convolution is defined as:

$$\tilde{L}^{(K)}_{gconv,t} = (W_k \circ \tilde{A^k})X_{\mathrm{t}} + B_{\mathrm{k}}, \tag{3}$$

where $\circ$ refer to element-wise matrix product, B is the bias and W a learnable matrix of weights. $\tilde{A^k}$ is defined as $min\{A^k + I, 1\}$.

In order to adapt this kind of networks to spatio-temporal environments, a LSTM layer is stacked with the convolutional one as with ConvLSTM in Sect. 3.2.2.

Note that we use nodes to represent the spatial measurement locations, which typically will be sensor stations or road segments, and edges to represent the spatial segments connecting those sensing locations. The adjacency matrix A which defines these spatial segments is usually built based on spatial metrics. For convenience and homogenization, we define for each dataset A as:

$$A_{i,j} = \begin{cases} 1 & (i,j \text{ are neighbors}) \\ 0 & \quad (\text{otherwise}) \end{cases} \tag{4}$$

where two locations i and j are considered neighbors if they are among the 4 closest areas without counting themselves. Through this definition of A, it is trivial to see that the convolution over each space zone makes use of information based on proximity.

3.3 Spatial agnosticism via convolutional networks

Now we show a series of spatial agnostic versions based on the models presented through Sect. 3.2 for spatio-temporal regression. These methods will help us to test the main hypothesis of this work: whether introducing spatial-adjacency bias is unquestionably the best option or not. For this purpose, each agnostic version needs to fulfill two requirements:

- No spatial information is introduced to the network.
- Past temporal information can be handled and introduced in the calculation of each new state.

By doing so, we will have several spatio-temporal methodologies that let us contrast our main premise.

As the objective of this work is not the specific development of models *per se*, in this section we propose a series of possibilities that meet the requirements defined to be spatially agnostic without limiting that other models of similar nature can be defined to test the same hypothesis.

3.3.1 Agnostic convolutional neural network (A-CNN)

To define an agnostic version of CNNs, we can work from Eq. 1. However, the kernel size is regularly used with equivalent values for its two dimensions $k_1 = k_2 = k$. In this case, not only this kernel uses different values for each component, but kernel size for spatial dimension must be equal to the number of spatial zones: $k_2 = S$. As a result, the convolution operation is made over all locations at once. The kernel size in the temporal dimension is defined as t_{past} and needs to be stipulated as part of the network architecture. An example of this kind of filter can be found in Fig. 3. The convolution itself writes as follows:

$$(x * K)(i,j) = \sum_{m}^{t_{\mathrm{past}}} \sum_{n}^{S} x(m,n)K(i-m,j-n) \tag{5}$$

The temporal dimension is dominated by a causal convolution. Generally, causal convolution ensures that the state created at time t derives only from inputs from time t to $t - t_{\mathrm{past}}$. In other words, it shifts the filter in the right temporal direction. Thus t_{past} can be interpreted as how many lags are been considered when processing an specific timestep. Given that previous temporal states are taken into account for each step and that parameters are shared all over the convolution, this methodology might be seen as some kind of memory mechanism by itself. Unlike memory-based RNNs (like LSTMs and GRUs) where the memory mechanism is integrated solely by learned via the hidden state, in this case t_{past} act as a variable that lets us take some control over this property.

In order to ensure that each input timestep has a corresponding new state when convolving, a padding of $P =$

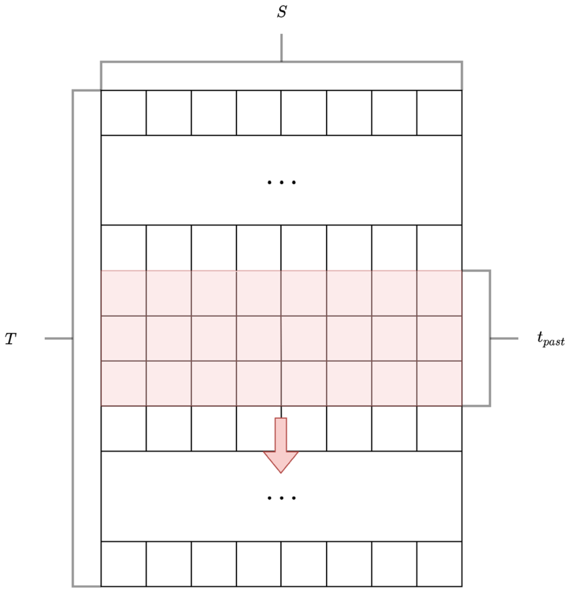


Fig. 3 Example of causal convolution spatially agnostic with $t_{\mathrm{past}} = 3$ through a spatio-temporal sequence of just one variable as defined previously

$t_{\text{past}} - 1$ at the top of the input “image” is required, and to guarantee temporal integrity, this padding must be done only at the top. By using convolution in this form, once the kernel has moved over the entire input image $T \times S$, the output image will be $T \times 1$. This process is summarized in Fig. 4.

Now, if we repeat this operation H times, we will create a new hidden state with H channels and output an image with $H \times T$ dimensions as the example in Fig. 5.

To give the network the opportunity to cover a spectrum of possibilities in terms of expressiveness as wide as a usual CNN for each channel, we simply use transposed convolution with a kernel size $k = (1, S)$ so the system can learn a $T \times S$ representation from a $T \times 1$ image. Figure 6 illustrates this idea.

Evidently, our new representation is usually composed by H hidden states, so this transposed convolution will use H filters. Finally, the complete procedure for an entire agnostic convolutional block is described graphically in Fig. 7.

Obviously, there are no restrictions with respect to the width dimension. For simplicity, we have considered convolutions in which only the number of channels is changed, meaning that images keep an $T \times S$ structure during all the computations. As we have described previously, this will help to normalize our experiments. However, as with CNNs, the dimensionality of hidden and output states might be different. Over all this process, the arrangement of the spatial dimension (columns) has no effect, meaning that it is not necessary to map the study areas in any specific way with the input of the network, as was done with the CNNs (Fig. 2).

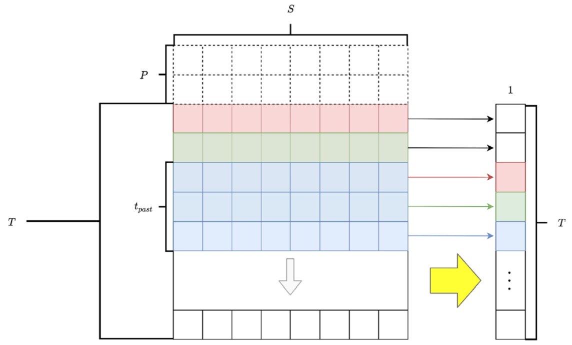


Fig. 4 Illustration of several step of the convolutional part of an agnostic convolutional block. After moving all over the input sequence, a $T \times 1$ image is produced. This new image compress information from all spatial locations and all input lags, keeping track of several of these ones in each convolution

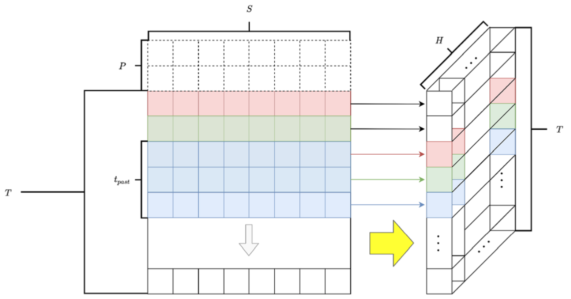


Fig. 5 By repeating operations described before, it is trivial to assemble hidden states as new channels in the latent sequence, meaning $T \times 1$ images with H channels

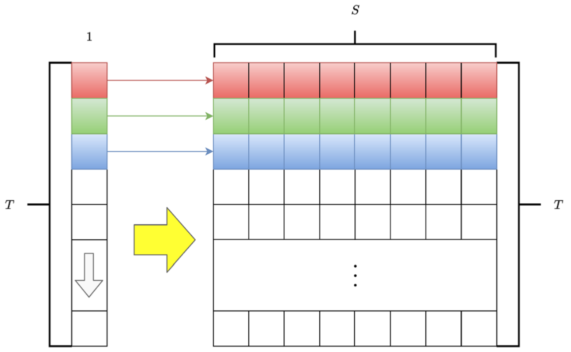


Fig. 6 Transposed convolution to produce a $T \times S$ images from $T \times 1$ latent sequences. Thanks to this process, we give the model same expressiveness opportunities as traditional CNNs

3.3.2 Agnostic ConvLSTM (A-ConvLSTM)

Once that the agnostic procedure for convolving has been presented in the previous section, the A-ConvLSTM is governed by Eq. 2 but changing the traditional convolution for this new approach (Sect. 3.3.1). The only difference with respect to A-CNN is that there is no need of causal convolution over the temporal dimension, as the LSTM module can handle it. Therefore, the input sequence lacks any spatial ordering procedure, letting us define this dimension arbitrarily.

3.3.3 Agnostic graph convolutional network (A-GCN-LSTM)

Following the structure of the GCN-LSTM presented in Sect. 3.2.3, its agnostic version is simply to define the adjacency matrix as the identity matrix: $A = I_S$. Thus, we can make sure that no spatial relation is being introduced or modeled explicitly. Otherwise, the network has the same functioning and characteristics as described before. Thus, the graph convolution takes the following form:

$$\tilde{L}^{(K)}_{gconv,t} = W_k X_t + B_k, \qquad (6)$$

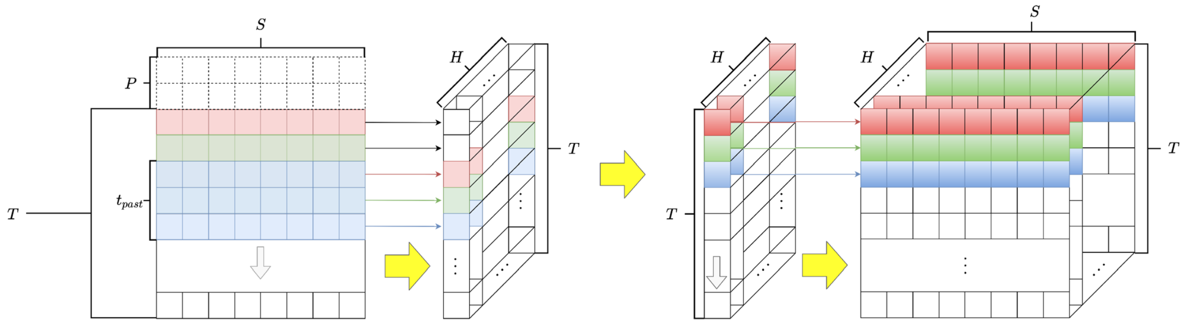


Fig. 7 Representation of a complete agnostic convolutional block. By assembling operations described before, from a $T \times S$ it is trivial to create hidden states capable of representing equivalent expressions compared to a traditional CNN, meaning $T \times S$ images with H channels

A comparison between each traditional model and its agnostic version is shown in Table 1. There it is summarized how each model formalizes *prior* information about the spatial dimension and how it affects their use.

3.4 Regressor block

Through Sects. 3.2 and 3.3, we have explored how to use convolution operations to learn a new hidden representation of the input sequence as an image with and without using *prior* spatial information or closeness assumptions. Now, in order to make a fair comparison between traditional networks and their respective agnostic versions, we have to carefully use this latent representation with $T \times S \times H$ dimensions (common to all models presented) to get a new $T' \times S$ predicted image. While this process can be done in multiple ways, it is desirable for this regressor block to fulfill several conditions:

(1) The same strategy has to be applicable to all models studied in this work.

(2) It can not explicitly share information between elements of the spatial dimension. This way, we make sure that space is only treated in the convolutional block of each model, and our results are not contaminated from other parts of the network.

(3) The number of parameters needs to be as low as possible and space-independent. Thus, we avoid overfitting or overinfluence problems.

(4) Lastly, although we have not found an option that is completely network architecture-independent (you can get a similar size of hidden dimension or total number of parameters, but not both), it is highly desirable that this regressor layer does not undergo too much variability between models.

A naive and simple approach would be using 1D convolutions after reshaping the $H \times T \times S$ image into a $(H \cdot T) \times S$, with $H \cdot T$ being the number of input channels. By convolving trough the spatial dimension with a kernel size of $k = 1$ and an output number of channels of T', we can be sure no information is shared through this dimension (2) and the number of parameters, which is $H \cdot T \cdot T'$, remains low compared to the complete network (3). Furthermore, all models that we will compare are based on a convolutional block which outputs an $H \times T \times S$, meaning that this regressor scheme can be applied to all of them, helping to standardize our experiments (1). As T is the same for all models and H never diverges more than one order of magnitude, we can be sure this layer has a similar impact for all cases (4). Figure 8 summarizes this block.

Although other options have been considered, as 2D convolutions and dense layers, they fail to meet some

Table 1 Summary of spatial treatment of each model

	Traditional	Agnostic
CNN	Shared kernel among locations. Ordering of spatial dimension based on closeness	One kernel for each location. No ordering needed
ConvLSTM	Shared kernel among locations. Ordering of spatial dimension based on closeness	One kernel for each location. No ordering needed
GCN-LSTM	Adjacency matrix defined by proximity	Identity matrix as adjacency matrix

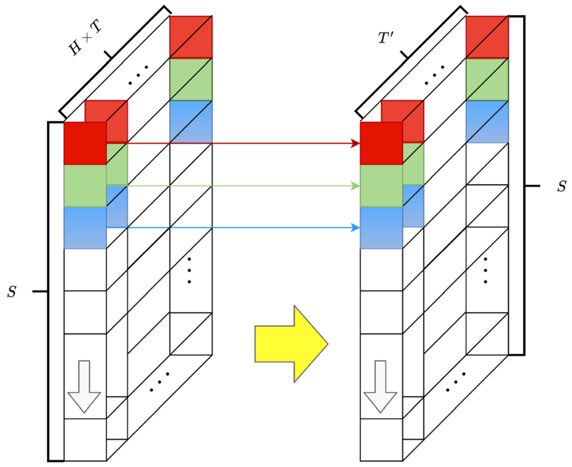


Fig. 8 Regressor block that satisfy applicability, spatial agnosticism and simplicity. By using a 1D convolution over the latent image $H \times T \times S$, we can produce a $T' \times S$ sequence that correspond to our forecast

conditions or need fine-tuning for each problem and model, making them less suitable for a fair comparison.

3.5 Temporal versus spatial distribution

Our work is based on the hypothesis that real spatio-temporal series might not share a similar behavior in their two dimensions. Even the well known fact that closer, spatially speaking, locations behaves similarly does not always suit well, meaning that the parameter sharing scheme of traditional CNNs might not be the best option. Concretely, when dealing with real problems, the system might have a high dependency on non-spatial phenomena, and data collection can have a great impact. As a result, closeness information can be lost or modified.

On the contrary, temporal information (or distribution) usually keeps the same structure for a wide range of problems. As air quality and mobility are high correlated to human being, the temporal pattern of this kind of series for each location tends to remain alike.

In order to prove our hypotheses, we will make use of statistical tools that characterize the aforementioned information.

3.5.1 Spatial dimension: Moran's *I*

According to [30], “Spatial autocorrelation or spatial dependence can be defined as a particular relationship between the spatial proximity among observational units and the numeric similarity among their values; positive spatial autocorrelation refers to situations in which the nearer the observational units, the more similar their values (and vice versa for its negative counterpart)... This feature violates the assumption of independent observations upon which many standard statistical treatments are predicated.” This property, which is precisely what we are interested in, can be measured by the well know Moran's I [31]. This test will let us quantify the degree of spatial autocorrelation existing in the different datasets that we will use between close locations taking into account this interdependency. As it is a test, Moran's I comes with a p-value which typifies statistical significance of the result. It is defined as:

$$I = \frac{S}{W} \frac{\sum_{\mathrm{i}} \sum_{\mathrm{j}} w_{\mathrm{ij}} (x_i - \bar{x})(x_j - \hat{x})}{\sum_i (x_i - \bar{x})^2} \tag{7}$$

where S is the number of spatial units indexed by i and j, x is the variable of interest, $\bar{x}$ is the mean of x, w_{ij} is a matrix of spatial weights based on neighbors, and W is the sum of all w_{ij}. As its value varies usually between -1 and $+1$, it is easily interpretable. Concretely, $+1$ implies similar values for close locations, 0 a random arrangement, and -1 opposite values.

As we also have a temporal dimension, we will average I for all timesteps. Through this test we want to compute solely spatial autocorrelation, without intervention of temporal relations between locations.

3.5.2 Temporal dimension: adaptative temporal dissimilarity measure

To compare the similarity between different time series (in our case, different spatial points), the same problem arises than with spatial autocorrelation: due to the interdependence relationship between measurements classical correlation index can not be applied. For example, Euclidean, Fréchet distances and Dynamic time warping are well known and widely used techniques when measuring time series similarity but do not handle the aforementioned issue well. To solve this problem, [32] proposed the Adaptative Temporal Dissimilarity Measure (ATDM) as an index that lets us measure the similarity between time series more robustly as it balances the proximity with respect to values and the proximity with respect to behavior. Ir writes as:

$$\mathrm{ATDM}(X_{T,i}, X_{T,j}) = f(\mathrm{cort}(X_{t,i}, X_{t,j})) \cdot \delta(X_{t,i}, X_{t,j}), \tag{8}$$

where δ references a classical distance (we will use Euclidean) and cort is

$$\mathrm{cort}(X_{T,i}, X_{T,j}) = \frac{\sum_t^{T-1} (X_{t+1,i} - X_{t,i})(X_{t+1,j} - X_{t,j})}{\sqrt{\sum_t^{T-1} (X_{t+1,i} - X_{t,i})^2} \sqrt{\sum_t^{T-1} (X_{t+1,j} - X_{t,j})^2}}. \tag{9}$$

Lastly, f writes as follow:

$$f(x) = \frac{2}{1 + \exp(kx)}, k \geq 0. \tag{10}$$

with this metric, the distance is squeezed into a coefficient in the interval (0, 2). When the correlation coefficient is 0, the ATDM is 1, and the correlation is not significant. When the correlation is positive, the value of the ATDM is less than 1; the more similar the two series are, the smaller the value is. On the contrary, the ATDM is more than 1 if the correlation is negative. The less similar the two series are, the larger the value is.

Thus, we can average the ATDM between all locations pairs for each spatio-temporal series. As this measure takes into account both values and behavior of the series, we can approximately get a global measure of temporal distribution similarity among points for each dataset.

When working with real data, in which depending on time granularity local properties of time series might be noisy, ATDM might not extract information correctly. In order to solve this, we compute an adjusted ATDM coefficient (ATDM$_{adj}$) which uses a smoother version of the input series as we are interested in global behavior of the temporal distribution. Concretely, we use moving average as it is simple and has shown to be a good approximator for time series. As moving average just smooths the series, we do not expect to corrupt the coefficient between series which are not really temporally correlated.

4 Experimental design

4.1 Data description

The different forecasting problems and the corresponding datasets are described below. Main dataset characteristics and statistics are provided in Table 2.

- *AcPol dataset* Provided by the Municipality of Madrid through its open data portal.[1] Acoustic pollution in Madrid in decibels, it measures equivalent continuous level with A frequency weighting, which is the assumed noise level constant and continuous over a period of time, corresponding to the same amount of energy than that actual variable level measured in the same period.
- *Beijing dataset* Presented by [33], it consists of traffic speed measurements for 15000 road segments recorded per minute. To make the traffic speed predictable for each road segment, it is aggregated via moving average in 15 minutes intervals. For this work, we select a subgroup of road segments spatially close.
- *BiciMad dataset* Supplied by EMT (Municipal Transport Company for its initials in Spanish) through its open data portal.[2] In this case, we tackle the bike sharing demand prediction by aggregating the overall number of bikes per station and timestep.
- *LOOP dataset* It contains data collected from inductive traffic loop detectors deployed on four connected freeways (I-5, I-405, I-90, and SR-520) in the Greater Seattle Area. It can be found in [34].
- *MATRA dataset* This dataset contains historical data of traffic measurements in the city of Madrid. The measurements are taken every 15 minutes at each point, including traffic intensity in number of cars per hour. Data are aggregated for each hour. While a dense and populated network of over 4.000 sensors is available, we decided to simplify and use only a selection of them. Available in the Municipality of Madrid open data portal (footnote 1).
- *METR-LA dataset* This dataset contains traffic information recapitulated from loop detectors in the highway of Los Angeles County. We use the partition provided by [17].
- *NO_2 dataset NO_2* in the city of Madrid. Hourly data for all measurement stations which include this pollutant. Available in the Municipality of Madrid open data portal (see footnote 1).
- *NYTaxi dataset* Provided by Taxi & Limousine Commission,[3] it consist of taxi trip location and duration in the city of New York. We focus our work on forecasting the number of taxi travels for each New York neighborhood with an average minimum number of one trip per day.
- *O3 dataset O_3* in the city of Madrid. Hourly data for all measurement stations which include this pollutant. Available in the Municipality of Madrid open data portal (see footnote 1).
- *PEMS-BAY dataset* This traffic dataset is collected by California Transportation Agencies (CalTrans) Performance Measurement System (PeMS). We use the partition provided by [17].

All datasets are Z-Score normalized by spatial point. We take as reference previous work as a criterion to choose T and T'. Thus, we can be sure of the plausibility of the results for all models. When no previous work is known, we use autocorrelation as a measurement of number of minimum lags (T) and focus only on a single timestep prediction ($T' = 1$).

[1] *Portal de datos abiertos del Ayuntamiento de Madrid*: https://datos.madrid.es/portal/site/egob/

[2] *Portal de datos abiertos EMT*: https://opendata.emtmadrid.es/Datos-estaticos/Datos-generales-(1)

[3] *NYCTaxi and Limousine Commission (TLC) Trip Record Data*: https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

Table 2 Details of data through experiments.

Dataset	Dates	Timestep	T	T'	S	Mean	Median	Std	ATDM	ATDM$_{adj}$	Moran's I
AcPol	2014/01/01–2019/03/31	1 day	7	1	30	56.8	60.2	15.1	0.36	0.36	0.03
Beijing	2017/01/04–2017/05/31	15 min	10	1	200	29.0	28.7	9.3	0.69	0.27	0.20
BiciMad	2019/01/01–2019/06/30	1 h	6	1	168	0	0	3.2	1.04	1.03	0.12
LOOP	2015/01/01–2015/03/31	5 min	10	1	323	57.2	60.6	11.8	0.84	0.47	0.31
MATR	2018/01/01–2019/12/31	1 h	24	6	120	445.5	254.8	539.6	5.6E-4	4.8E-4	0.09
METR-LA	2012/03/01–2012/06/30	5 min	12	3	207	53.4	62.3	20.6	0.02	0.02	0.24
NO2	2017/01/01–2019/12/31	1 h	48	48	24	37.5	29	28.9	0.04	0.0	0.13
NYTaxi	2016/01/01–2016/06/30	1 h	6	1	70	4.8	0	11.3	0.55	0.34	0.24
O3	2017/01/01–2019/12/31	1 h	48	48	14	50.6	50	34.3	0.03	0.0	0.11
PEMS-BAY	2017/01/01–2017/05/31	5 min	12	3	325	62.6	65.3	9.6	0.64	0.15	0.23

Dates reflects starting and ending points of data, *Timestep* corresponds to the duration of one timestep. T, T', and S were defined in Sect. 3.1 as input timesteps, output timesteps, and number of spatial locations. *Mean*, *Median*, and *Std* condense main data statistics. *ATDM*, *ATDM*$_{\text{adj}}$, and *Moran's I* summarize information about spatial and temporal distribution similarity between locations

From Table 2, we can see how our chosen datasets cover a wide range of spatio-temporal circumstances and the high variety and variability of data. Also, our main hypotheses are confirmed: Moran's I show a clear no-spatial autocorrelation pattern for our series, and although not completely uncorrelated, most series are close to 0. All p-values are lower than 0.05. It is worth noting as proof of plausibility for these values that [22] computed the coefficient I for the complete Beijing traffic dataset at some hours, reporting a similar value to ours. ATDM values tend to be low, which is representative of similar temporal distributions in the datasets. As we expected, ATDM$_{adj}$ represents better this idea. Datasets with a clear temporal pattern but locally noisy, as Beijing, LOOP, and PEMS-BAY, are better described by this coefficient.

Given that spatial locations are by default in arbitrary order, it is necessary to sort and structure them in order to fully exploit spatial information with traditional models. By computing a hierarchical tree (dendrogram) using an agglomerative hierarchical clustering algorithm and traversing recursively the tree, it is possible to approximately sort the points by distance.

4.2 Architecture models

We compare agnostic models with widely used spatio-temporal series regression models based on the convolution operator. Details concerning its architectures are:

- *A-CNN* Through the process batch normalization and ReLU activation function are used.
- *CNN* A standard CNN followed by a batch normalization layer and ReLU activation function. It uses a 3×3 kernel.
- *A-ConvLSTM* ReLU activation function after convolution. No batch normalization.
- *ConvLSTM* A standard ConvLSTM that uses a 3×3 kernel. ReLU activation function after convolution. No batch normalization.
- *A-GCN-LSTM* ReLU activation function.
- *GCN-LSTM* A classical approach for GCN which lets us exploit explicitly information from the k-hop (k-th order) neighborhood of each node in the graph. In our experiments, we set $k = 3$ and use ReLU activation function.

As we are interested in deepening in how the convolution operator and the spatial dimension are related, we do not include any RNN or FNN-based approach.

4.3 Experimental design

In order to make a comparison as fair as possible, we decided to proceed with all models as follows:

- They will consist uniquely in a convolutional layer and a regressor layer. For all of them, the convolutional layer will enrich input information by constructing a $H \times T \times S$ image from a $T \times S$ sequence as described in Sect. 3.1.
- The regressor layer consists of a 1D convolution, as explained in Sect. 3.4. Thus, we make sure no model is taking advantage or exploiting further spatial information.
- The number of parameters in the convolutional layer need to remain similar and in the same magnitude order. Given regressor layer's architecture and the fact that it is the same for all models, we expect that this is enough to eliminate possible bias.

- A weight decay (L2 regularization) of 10^{-3} is used to prevent overfitting.

Some other minor details are that all the models are trained using the mean squared error (MSE) as objective function with the RMSprop optimizer, as it has shown good performance in non-stationary scenarios. Batch size is 256, momentum is set to 0.9, the initial learning rate is 0.001, and both early stopping and learning rate decay are implemented in order to avoid overfitting and improve performance. The experiments are run in a NVIDIA RTX 2070.

As we have standardized the experiments, no hyperparameter tuning is needed in general. Solely t_{past} for A-CNN needs to be adjusted, which will be tuned via standard grid search.

4.4 Validation scheme

As stated in [35], standard *k*-cross-validation is the way to go when validating neural networks for time series if several conditions are met. Specifically, that we are modeling a stationary nonlinear process, that we can ensure that the leave-one-out estimation is a consistent estimator for our predictions and that we have serially uncorrelated errors.

While the first and the third conditions are trivially fulfilled for our problem, the second one needs to be specifically treated. Given that some input sequences might share elements among different sets(training, validation, and test), *prior* information could be entangled leading to data leakage. Due to this problem, it is not possible to create random folds, and it is necessary to specify a separation border among previously defined sets. Particularly, a 10-cross-validation scheme without repetition is used during all experiments, with a 80/10/10% scheme for train/validation/test sets for each fold.

4.5 Error metrics

To evaluate the precision of each model, we computed root mean squared error (RMSE) and bias. In a spatio-temporal context [36], they are defined as:

$$\text{RMSE} = \sqrt{\frac{1}{T'S}\sum_{i=1}^{T'}\sum_{i=1}^{S}(\tilde{x}_{t'_i,s_j} - x_{t'_i,s_j})^2}, \tag{11}$$

$$\text{bias} = \frac{1}{T'S}\sum_{i=1}^{T'}\sum_{j=1}^{S}(\tilde{x}_{t'_i,s_j} - x_{t'_i,s_j}), \tag{12}$$

For all these metrics, the closer to zero they are the better the performance is. While RMSE already provides a dispersion measure respect to real series, bias is better to find particular predispositions when making predictions.

5 Results

A general comparison of the different error metrics for all models can be seen in Table 3.

To better visualize the error over all datasets, Fig. 9 shows RMSE distribution. From this figure, we can deduct that, in general terms, agnostic models show a similar behavior than their respective main competitors.

In order to inquire into these results and provide statistical evidence, a Friedman rank test was performed over the error distribution for all datasets. A Friedman statistic of $F = 21.6$, distributed according to a χ^2 with 5 degrees of freedom obtains a p-value of $6.2e-4$ with $\alpha = 0.05$, which provides evidence of the existence of a significant difference between the algorithm.

Given that Friedman's null hypothesis was rejected, a post-hoc pairwise non-parametric-based comparison was carried out to check the differences between the proposed algorithms with Holm and Benjamini-Hochberg adjustments. As we are especially interested in testing whether the introduction of spatial information as a *prior* is necessary or not, Table 4 shows statistical significance in the traditional-agnostic model comparison for all datasets. Through these tests we compare if there are significant differences between the means of two different algorithms error distributions. Thus, for each hypothesis the test accepts or rejects the idea that the two models that compose the hypothesis generate, statistically speaking, the same error distributions. By looking at this table we can confirm our initial claim since there is not enough evidence to support that traditional methods suppose an improvement over their agnostic versions. In fact, the only comparison that yields a significant result (hypothesis I) show evidence in favor of the agnostic model.

In terms of computational performance, Table 5 summarizes average run times per fold, model, and dataset, and the number of parameters per dataset for all models (recall that, to facilitate a fairer comparison, all models have the same number of parameters for every problem, see Sect. 4.3). Again, no differences are reported between traditional and agnostic models neither. As we would expect, A-CNN and CNN models show a great advantage in terms of time consumption compared to the rest of the methodologies.

To further validate one of the most important statements of this work, i.e., to ensure that the models we have presented as spatially agnostic really are, we propose to randomly permutate the spatial dimension of data before training. As we just want to compare the behavior of the

Table 3 Average performance per model and dataset. For a more detailed view of error metrics distribution, see Fig. 9

	AcPol		Beijing		BiciMad		LOOP		MATR	
	RMSE	Bias	RMSE	Bias	RMSE	Bias	RMSE	Bias	RMSE	Bias
A-CNN	7.06	0.22	2.74	0.02	2.75	− 2.0E-4	5.06	− 0.07	115.65	− 5.30
CNN	8.52	− 0.04	4.52	− 0.02	2.94	− 0.01	4.59	0.05	141.99	0.13
A-ConvLSTM	6.22	0.04	2.64	7.3E-3	2.74	− 8.3E-3	4.52	0.02	111.74	− 0.06
ConvLSTM	5.46	− 3.0E-3	2.26	− 0.15	2.89	− 0.01	3.71	0.03	115.29	− 2.85
A-GCN-LSTM	7.45	0.17	2.88	0.02	2.76	6.7E-4	5.77	− 0.53	136.48	2.97
GCN-LSTM	8.01	0.03	2.76	0.09	2.70	6.2E-3	5.02	− 0.18	132.14	0.43

	METR-LA		NO2		NYTaxi		O3		PEMS-BAY	
	RMSE	Bias	RMSE	Bias	RMSE	Bias	RMSE	Bias	RMSE	Bias
A-CNN	9.52	0.29	23.17	0.27	2.97	0.02	22.13	1.51	3.98	− 0.03
CNN	10.00	− 0.01	24.26	− 0.03	3.52	0.06	23.30	0.607	3.98	0.03
A-ConvLSTM	9.13	0.24	22.79	0.01	2.86	1.3E-4	21.56	1.00	3.66	− 0.03
ConvLSTM	7.86	− 0.05	24.25	0.31	3.16	− 0.01	21.56	0.71	2.42	0.04
A-GCN-LSTM	10.14	0.14	23.51	0.23	2.87	2.7E-3	22.71	0.25	4.16	0.14
GCN-LSTM	10.17	− 0.58	24.98	− 0.90	2.88	0.01	23.39	1.75	4.07	− 0.68

Table 4 Adjusted Holm and Benjamini-Hochberg *p*-values with pairwise rejected hypothesis at $\alpha = 0.05$ for all datasets

i	hypotheses	$p_{\text{unajusted}}$	p_{holm}	p_{BH}
I	A-CNN versus CNN	0.014	0.048	0.023
II	A-ConvLSTM versus ConvLSTM	0.77	1	0.825
III	A-GCN-LSTM versus GCN-LSTM	0.736	1	0.846

A *p*-value lower than α suggest that both algorithms produce different error distributions

Table 5 Average run time per fold in seconds and approximate number of parameters used per dataset

	AcPol	Beijing	BiciMad	LOOP	MATR	METR-LA	NO2	NYTaxi	O3	PEMS-BAY	Average
A-CNN	1.0	16.7	4.1	97.7	26.7	36.5	56.8	3.6	42.8	67.9	36.3
CNN	1.4	8.1	13.4	117.5	44.5	68.6	33.2	18.4	22.5	48.0	37.8
A-ConvLSTM	4.8	41.9	5.1	229.1	411.8	179.6	67.8	9.2	56.2	349.0	135.5
ConvLSTM	2.6	103.2	38.2	350.0	422.5	238.7	74.8	13.0	56.8	400.7	171.6
A-GCN-LSTM	18.1	72.6	15.0	168.7	146.0	83.7	449.4	24.4	221.2	172.3	137.1
GCN-LSTM	13.3	71.9	12.5	190.5	111.7	77.9	467.4	30.5	221.6	199.9	141.0
Number of parameters	∼ 50K	∼ 200K	∼ 150K	∼ 250K	∼ 200K	∼ 150K	∼ 200K	∼ 150K	∼ 150K	∼ 250K	

different methods when input data is not sorted, we are only interested in studying how the error distributions are modified when this perturbation is introduced in the system, and not in pure performance. Given that ConvLSTM and A-ConvLSTM have shown to be a statistically significant better option than the other models, we will use only these two models in this experiment.

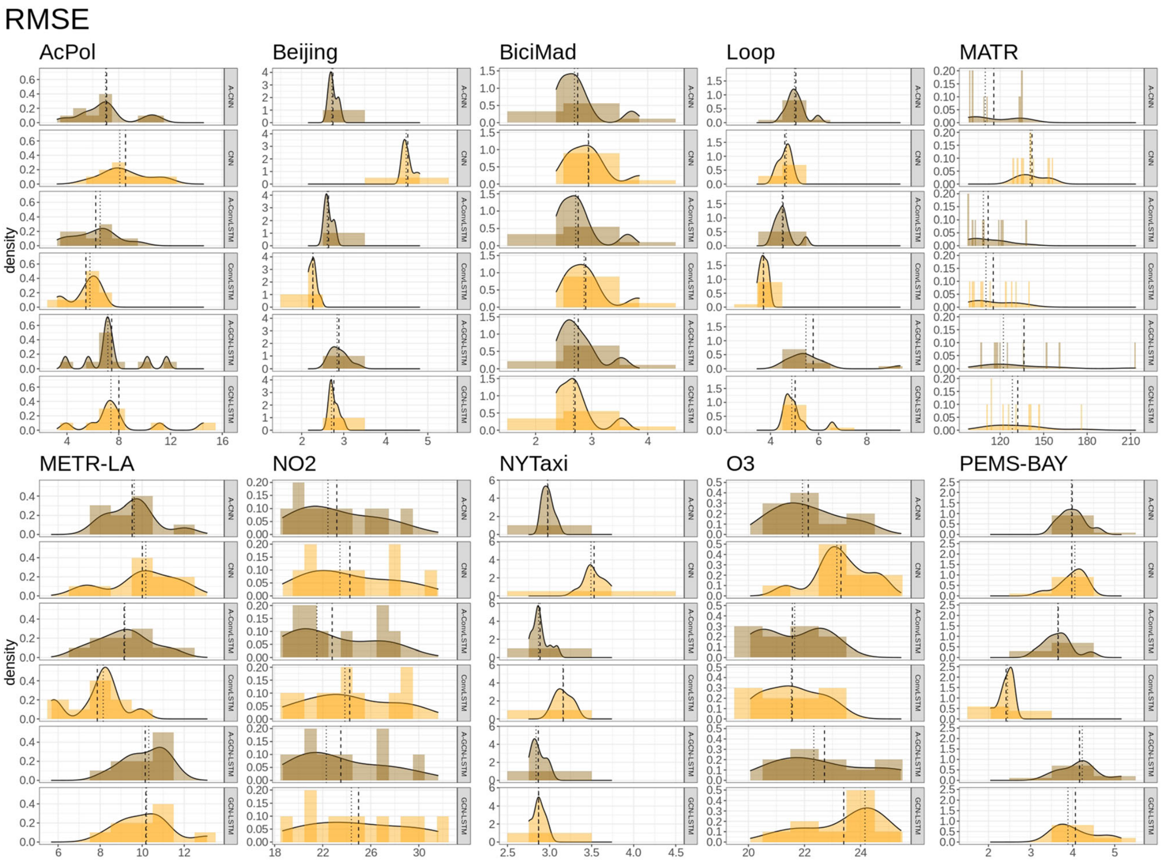


Fig. 9 RMSE distribution for each model and dataset, dashed vertical line represents the mean, dotted vertical line represents median

In Fig. 10 we can visualize the RMSE results for both models before and after (*model name-perm*) the random permutation.

From this last figure we can clearly see that error distributions for A-ConvLSTM and A-ConvLSTM-perm are practically identical for all the problems, while that does not happen for ConvLSTM and ConvLSTM-perm. Thus, we carry out a post-hoc pairwise non-parametric-based comparison to check the differences between the models with Holm and Benjamini-Hochberg adjustments. Table 6 shows the aforementioned *p*-values, marking with asterisks (*) those that are statistically significant. In the table, "A-ConvLSTM" refers to the comparison A-ConvLSTM vs A-ConvLSTM-perm, while "ConvLSTM" refers to the comparison ConvLSTM vs ConvLSTM-perm.

6 Discussion

First of all, we can verify the goodness of our experiments presented in Table 3 by direct comparison with analogous studies [16, 17, 33, 34, 37], showing that our results are in line with them. Since most of the datasets have already been used, we can extrapolate this idea to those which have not.

The performance of the different strategies over individual datasets is directly associated with the spatial autocorrelation metric in Table 2. On the one hand, datasets with a higher value of Moran's *I* have a propensity to show better performance with traditional models (Beijing, Loop, METR-LA, and PEMS-BAY). On the other hand, datasets with lower values of the same metric usually show better behavior with the agnostics versions (AcPol, BiciMad, MATR, NO2, NYTaxi, and O3).

Regarding to permuting the input tensors in their spatial dimension in order to know the impact of this ordering on the performance of each type of model, Table 6 lets us

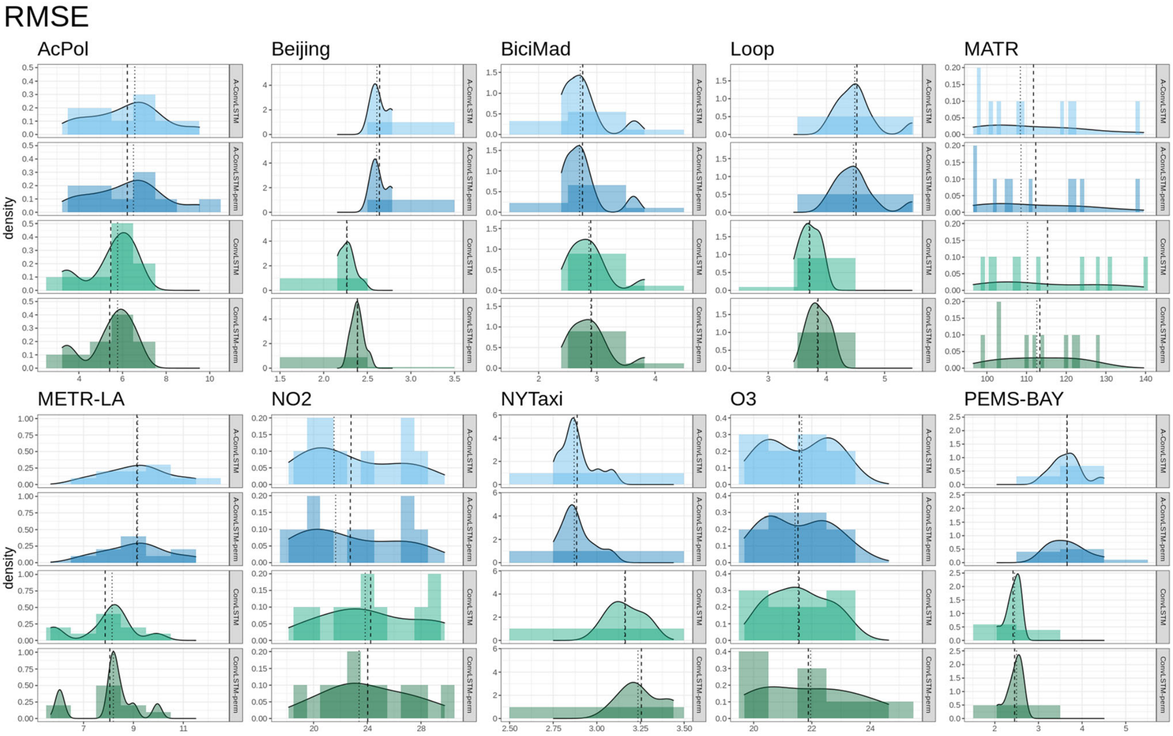


Fig. 10 RMSE distribution for each model and dataset before and after training with random permutations in their spatial dimension. Dashed vertical line represents the mean, dotted vertical line represents median. In blue, A-ConvLSTM-based models and in green ConvLSTM-based models (Color figure online)

conclude that A-ConvLSTM shows spatial agnosticism, and its performance is unaffected by how the spatial dimension is treated. However, the ConvLSTM presents an important discrepancy in terms of performance when unsorting the grid. Although this premise holds in general terms over all datasets, it can be seen again that the results are directly related to correlation metrics in Table 2: those datasets with a higher value of Moran's *I* tend to suffer more with the permutation test (Beijing, LOOP, METR-LA, and PEMS-BAY). As in those cases the spatial autocorrelation is higher, sharing parameters in the spatial dimension is more beneficial, and changing the grid has a greater effect.

As pointed out by reviewer #3, from a theoretical point of view, spatial dimension presents a non dominant pattern while temporal dimension generates smooth and similar fluctuations between all locations. Given that convolution operations have a tendency to average out the close patterns, this spatial non dominant relationships might not be evident after the convolutional operations. As a result, neural network's learning barely depends on the spatial relationships of adjacency and proximity, as shown by our results.

Finally, and given our results, we can provide some guidelines in order to help other practitioners working with real spatio-temporal problems:

- Assuming neighborhood-based relations as a premise when approaching a spatio-temporal problem with neural networks might not always be the best option. Instead of naively assuming these spatial relations, it might be beneficial to dig more deeply in the data analysis or to rethink how the problem is addressed.

  Concretely, real datasets do not necessarily are similar through spatial locations, contrary to what is usually assumed. Thus, the nature of data should be reflected when defining the network architecture. In any case, further considerations should be given to preliminary studies of the spatial distribution of the data.
- When the distribution of the data shows a clear spatial relationship based on neighborhood, as in the case of large traffic sensor networks, the traditional format of convolution-based networks might be advantageous. However, when this is not clearly verified, as for example with air quality, models do not show improvement by sharing weights between different locations.

Table 6 Adjusted Holm and Benjamini-Hochberg p-values with pairwise rejected hypothesis at $\alpha = 0.05$ for all datasets after testing spatial agnosticism via random permutation

	AcPol			Beijing			BiciMad			LOOP			MATR		
	$p_{\text{unajusted}}$	p_{holm}	p_{BH}	$p_{\text{unajusted}}$	p_{holm}	p_{BH}	$p_{\text{unajusted}}$	p_{holm}	p_{BH}	$p_{\text{unajusted}}$	p_{holm}	p_{BH}	$p_{\text{unajusted}}$	p_{holm}	p_{BH}
A-ConvLSTM	0.922	1	0.922	0.193	0.193	0.193	0.496	0.496	0.496	0.027*	0.027*	0.027*	0.232	1	0.668
ConvLSTM	0.922	1	0.922	0.014*	0.027*	0.016*	0.027*	0.027*	0.027*	0.002*	0.012*	0.002*	0.557	1	0.668
	METR-LA			NO2			NYTaxi			O3			PEMS-BAY		
	$p_{\text{unajusted}}$	p_{holm}	p_{BH}	$p_{\text{unajusted}}$	p_{holm}	p_{BH}	$p_{\text{unajusted}}$	p_{holm}	p_{BH}	$p_{\text{unajusted}}$	p_{holm}	p_{BH}	$p_{\text{unajusted}}$	p_{holm}	p_{BH}
A-ConvLSTM	0.777	0.777	0.777	0.777	0.984	0.777	0.375	0.375	0.375	0.557	1	1	0.846	0.846	0.846
ConvLSTM	0.011*	0.02*	0.012*	0.492	0.984	0.59	0.01*	0.022*	0.013*	0.846	1	0.846	0.004*	0.012*	0.005*

Rejected hypothesis (meaning both algorithms produce different error distributions) are marked with *

- If there is not enough available evidence about the spatial distribution characteristics, spatially agnostic models might be best suited as they are capable of performing well while being less laborious to work with.
- In any case, consider using spatial agnostic models if your needs in terms of precision must be balanced with the available resources.

7 Conclusions

Through this work, we have explored how classical spatial assumptions based on closeness are not always the best deal when working with convolutional neural networks for spatio-temporal series regression. Due to their usual lack of spatial autocorrelation, other alternatives might be more suited. In order to test this idea, we have compared several versions of convolutional-based models that make no use of *prior* spatial information (neither directly nor indirectly), namely spatial agnostic, with their respective traditional forms. Spatial agnostic models are a perfect tool to contrast our hypothesis as they do not use extra modules or steps as others but tackle the problem directly purely via convolutions.

After extensive and standardized experimentation, we can confirm our main hypothesis: the inclusion of adjacency-based representations of the spatial distribution of real data does not necessarily fit well for the classical convolutional shared-weights scheme. Concretely, without using any specific spatial mechanism, spatial agnostic models have been shown to be equal in performance to some of the most notable spatio-temporal models. Also, we have shown how these models, unlike traditional convolutional methods, are really spatially agnostic, and how this is related to the spatial autocorrelation of the series. Furthermore, beyond proving our initial hypothesis we have shown how our methodology is simpler and less laborious to work with, offering the possibility of obtaining good performance without having to carry out extra research about the application domain. Finally, by analyzing ten different datasets with different spatio-temporal conditions each, we can confirm the statistical significance of these statements with a confidence of 95%.

Some directions for future work include using agnostic versions of convolutional-based networks in those fields where they show clear benefits (such as pollution forecasting). Also, our work is a good starting point to rethink the way of working with spatio-temporal series if we want to extract and make use of the spatial information of the problem more efficiently and simply beyond classical adjacency hypothesis. Finally, it would be of great interest to propose and extend the same research questions to other learning algorithms beyond neural networks.

Acknowledgements This research has been partially funded by the *Empresa Municipal de Transportes* (EMT) of Madrid under the chair *Aula Universitaria EMT/UNED de Calidad del Aire y Movilidad Sostenible*.The authors are grateful to the anonymous reviewers for their constructive and helpful suggestions and comments. Especially to reviewer # 3, whose suggestions have directly contributed to the theoretical foundations of the article.

Funding Partially funded by the *Empresa Municipal de Transportes* (EMT) of Madrid.

Data Availibility specified through the manuscript.

Code availability https://github.com/rdemedrano/SANN

Declarations

Conflict of interest None.

References

1. Leung Y, Zhou Y, Lam KY, Fung T, Cheung KY, Kim T, Jung H (2019) Integration of air pollution data collected by mobile sensors and ground-based stations to derive a spatiotemporal air pollution profile of a city. Int J Geograph Inf Sci 33(11):2218–2240. https://doi.org/10.1080/13658816.2019.1633468
2. LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. Neural comput 1(4):541–551
3. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ (eds) Advances in neural information processing systems. Curran Associates, Inc., New York
4. Taigman Y, Yang M, Ranzato M, Wolf L (2014) DeepFace: closing the gap to human-level performance in face verification. IEEE Conf Comput Vis Pattern Recognit. https://doi.org/10.1109/CVPR.2014.220
5. Zhao B, Lu H, Chen S, Liu J, Wu D (2017) Convolutional neural networks for time series classification. J Syst Eng Electron 28(1):162–169. https://doi.org/10.21629/JSEE.2017.01.18
6. Cui Z, Chen W, Chen Y (2016) Multi-scale convolutional neural networks for time series classification. arXiv:1603.06995 [cs] . ArXiv: 1603.06995
7. Rodrigues F, Pereira FC (2020) Beyond expectation: deep joint mean and quantile regression for spatiotemporal problems. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2020.2966745
8. Tu E, Kasabov N, Yang J (2017) Mapping temporal variables into the neucube for improved pattern recognition, predictive modeling, and understanding of stream data. IEEE Trans Neural Netw Learn Syst 28(6):1305–1317. https://doi.org/10.1109/TNNLS.2016.2536742
9. Nam H, Han B (2016) Learning multi-domain convolutional neural networks for visual tracking. pp. 4293–4302
10. Kappeler A, Yoo S, Dai Q, Katsaggelos AK (2016) Video super-resolution with convolutional neural networks. IEEE Trans Comput Imag 2(2):109–122. https://doi.org/10.1109/TCI.2016.2532323
11. Liu Z, Li Z, Wang R, Zong M, Ji W (2020) Spatiotemporal saliency-based multi-stream networks with attention-aware LSTM for action recognition. Neural Comput Appl 32(18):14593–14602. https://doi.org/10.1007/s00521-020-05144-7
12. Jo D, Yu B, Jeon H, Sohn K (2019) Image-to-image learning to predict traffic speeds by considering area-wide spatio-temporal dependencies. IEEE Trans Veh Technol 68(2):1188–1197. https://doi.org/10.1109/TVT.2018.2885366
13. Guo S, Lin Y, Li S, Chen Z, Wan H (2019) Deep spatial-temporal 3D convolutional neural networks for traffic data forecasting. IEEE Trans Intell Transp Syst 20(10):3913–3926. https://doi.org/10.1109/TITS.2019.2906365
14. Ai Y, Li Z, Gan M, Zhang Y, Yu D, Chen W, Ju Y (2019) A deep learning approach on short-term spatiotemporal distribution forecasting of dockless bike-sharing system. Neural Comput Appl 31(5):1665–1677. https://doi.org/10.1007/s00521-018-3470-9
15. Wu Y, Tan H, Qin L, Ran B, Jiang Z (2018) A hybrid deep learning based traffic flow prediction method and its understanding. Transp Res Part C Emerg Technol 90:166–180. https://doi.org/10.1016/j.trc.2018.03.001
16. de Medrano R, Aznarte JL (2020) A spatio-temporal attention-based spot-forecasting framework for urban traffic prediction. Appl Soft Comput 96:106615. https://doi.org/10.1016/j.asoc.2020.106615
17. Li Y, Yu R, Shahabi C, Liu Y (2018) Diffusion convolutional recurrent neural network: data-driven traffic forecasting. arXiv:1707.01926 [cs, stat] . ArXiv: 1707.01926
18. Guo S, Lin Y, Feng N, Song C, Wan H (2019) Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the AAAI conference on artificial intelligence 33(01): 922–929. https://doi.org/10.1609/aaai.v33i01.3301922
19. Zhang Y, Cheng T (2020) Graph deep learning model for network-based predictive hotspot mapping of sparse spatio-temporal events. Comput Environ Urban Syst 79:101403. https://doi.org/10.1016/j.compenvurbsys.2019.101403
20. Zhang Y, Cheng T, Ren Y, Xie K (2020) A novel residual graph convolution deep learning model for short-term network-based traffic forecasting. Int J Geograph Inf Sci. https://doi.org/10.1080/13658816.2019.1697879
21. Zhou Z, Li X (2017) Graph convolution: a high-order and adaptive approach. arXiv:1706.09916 [cs, stat] . ArXiv: 1706.09916
22. Lu F, Liu K, Duan Y, Cheng S, Du F (2018) Modeling the heterogeneous traffic correlations in urban road systems using traffic-enhanced community detection approach. Phys A Stat Mech Appl 501:227–237. https://doi.org/10.1016/j.physa.2018.02.062
23. Asadi R, Regan AC (2020) A spatio-temporal decomposition based deep neural network for time series forecasting. Appl Soft Comput 87:105963. https://doi.org/10.1016/j.asoc.2019.105963
24. Aram P, Kadirkamanathan V, Anderson SR (2015) Spatiotemporal system identification with continuous spatial maps and sparse estimation. IEE Trans Neural Netw Learn Syst 26(11):2978–2983. https://doi.org/10.1109/TNNLS.2015.2392563
25. Do LNN, Vu HL, Vo BQ, Liu Z, Phung D (2019) An effective spatial-temporal attention based neural network for traffic flow prediction. Transp Res Part C Emerg Technol 108:12–28. https://doi.org/10.1016/j.trc.2019.09.008
26. Yu B, Lee Y, Sohn K (2020) Forecasting road traffic speeds by considering area-wide spatio-temporal dependencies based on a graph convolutional neural network (GCN). Transp Res Part C Emerg Technoloies 114:189–204. https://doi.org/10.1016/j.trc.2020.02.013

27. Wu Z, Pan S, Long G, Jiang J, Chang X, Zhang C (2020) Connecting the dots: multivariate time series forecasting with graph neural networks. KDD 2020 . ArXiv: 2005.11650
28. Uselis A, Lukoševičius M, Stasytis L (2020) Localized convolutional neural networks for geospatial wind forecasting. arXiv: 2005.05930 [cs, stat] . ArXiv: 2005.05930
29. Shi X, Chen Z, Wang H, Yeung DY, Wong WK, Woo WC (2015) Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In: Cortes C, Lawrence ND, Lee DD, Sugiyama M, Garnett R (eds) Advances in neural information processing systems. Curran Associates, Inc., New York
30. Lee SI (2017) Correlation and spatial autocorrelation. Springer, Berlin
31. Moran PAP (1950) Notes on continuous stochastic phenomena. Biometrika 37(1/2):17–23
32. Chouakria AD, Nagabhushan PN (2007) Adaptive dissimilarity index for measuring time series proximity. Adv Data Anal Classif 1(1):5–21. https://doi.org/10.1007/s11634-006-0004-6
33. Liao B, Zhang J, Wu C, McIlwraith D, Chen T, Yang S, Guo Y, Wu F (2018) Deep sequence learning with auxiliary information for traffic prediction. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery and data mining. ACM
34. Cui Z, Henrickson K, Ke R, Wang Y (2019) Traffic graph convolutional recurrent neural network: a deep learning framework for network-scale traffic learning and forecasting. IEEE Trans Intell Transp Syst. https://doi.org/10.1109/TITS.2019.2950416
35. Bergmeir C, Hyndman RJ, Koo B (2018) A note on the validity of cross-validation for evaluating autoregressive time series prediction. Comput Stat Data Anal 120:70–83. https://doi.org/10.1016/j.csda.2017.11.003
36. Wikle C.K, Zammit-Mangion A, Cressie, N (2019) Spatio-temporal statistics with R, 1 edn. Chapman and Hall/CRC, Boca Raton, Florida : CRC Press, [2019] . https://doi.org/10.1201/9781351769723. https://www.taylorfrancis.com/books/9780429649783
37. Navares R, Aznarte JL (2020) Predicting air quality with deep learning LSTM: towards comprehensive models. Ecol Inf 55:101019. https://doi.org/10.1016/j.ecoinf.2019.101019

Chapter 5

SOCAIRE: Forecasting and monitoring urban air quality in Madrid

Type:	Published article
Journal:	Environmental Modelling & Software
Authors:	Rodrigo de Medrano & José Luis Aznarte
Published:	May 2021
Impact factor:	4.807
5-Year Impact factor:	5.317
Quartile:	Q1 (Computer Science - Interdisciplinary applications)
DOI:	10.1016/j.envsoft.2021.105084
Contribution:	Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing.

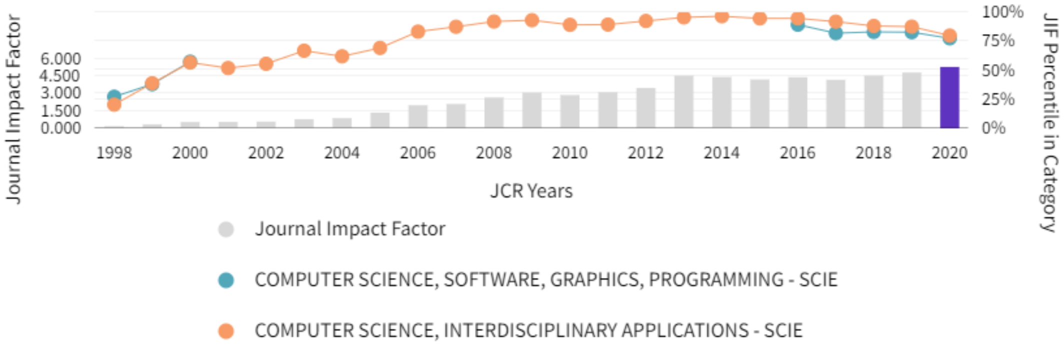


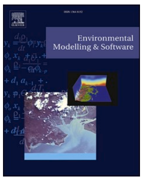
FIGURE 5.1: Impact factor: Environmental Modelling & Software.

Environmental Modelling and Software 143 (2021) 105084

Contents lists available at ScienceDirect

Environmental Modelling and Software

journal homepage: www.elsevier.com/locate/envsoft



SOCAIRE: Forecasting and monitoring urban air quality in Madrid

Rodrigo de Medrano [a], Víctor de Buen Remiro [b], José L. Aznarte [a,*]

[a] *Artificial Intelligence Department, Universidad Nacional de Educación a Distancia — UNED, Madrid, Spain*
[b] *Inverence, Madrid, Spain*

ARTICLE INFO

Keywords:
Air quality
Spatio-temporal series
Statistical modeling
Neural networks

ABSTRACT

Air quality has become a central issue in public health and urban planning management, due to the proven adverse effects of airborne pollutants. Considering temporary mobility restriction measures used to face low air quality episodes, the capability of foreseeing pollutant concentrations is crucial. We thus present SOCAIRE (Spanish acronim for "operational forecast system for air quality"), an operational tool based on a Bayesian and spatiotemporal ensemble of neural and statistical nested models. SOCAIRE integrates endogenous and exogenous information in order to predict and monitor future distributions of the concentration for the main pollutants. It focuses on modeling available components which affect air quality: past concentrations of pollutants, human activity, and numerical pollution and weather predictions. This tool is currently in operation in Madrid, producing daily air quality predictions for the next 48 h and anticipating the probability of the activation of the measures included in the city's official air quality NO_2 protocols through probabilistic inferences about compound events.

1. Introduction

During the last decades, an increasing number of studies point out that degraded air quality is a major problem in cities around the world (Martuzzi et al., 2006; Héroux et al., 2015). While there is general consensus that it causes health problems (Kim et al., 2015a; Özkaynak et al., 2009), how dangerous it can be is still a matter of debate (Sellier et al., 2014). Even in the best case scenario, this seemingly endemic issue affecting the life in big cities is already considered one of the main causes of both direct and indirect mortality (Badyda et al., 2017).

Of all the tools and systems that help fight pollution, the prediction of future pollutant concentrations or levels is of principal importance for air quality management and control (Bai et al., 2018). Air quality forecasting systems allow for sending out warnings of upcoming high pollution episodes to the population in the short-term, so that appropriate measures can be taken to minimize as far as possible the damage caused by these episodes. As an example, the city of Madrid, in order to comply with European regulations European Union, 2008, devised an air quality protocol which includes restrictions to the use of polluting vehicles when the concentrations of NO_2 reach certain thresholds. Consequently, foreseeing the activation of such restrictions is critical both for the decision makers (which need to announce them in advance) and for the vehicle owners (which need to plan their transport alternatives).

The use of data-driven approaches to predict and control air quality is not new. Following the discussion started in Breiman (2001), when approaching a modeling problem two families of methodologies (or "cultures", in Breiman terms) coexist: the data modeling culture, based on the search for a stochastic data model (for example, time difference equations, in the case of air quality) that capture the inner behavior of the intervening physical processes, and the algorithmic modeling culture, based on the use of algorithms to directly learn the model from data. Given that pollutant concentrations can be seen as time series, the stochastic data modeling usually deals with using ARMA based methods (Kumar and Jain, 2010; Hassanzadeh et al., 2009). However, this kind of models have trouble handling high non-linearities and high dimensional environments. To solve this problem, and thanks to the big amount of data that is gathered nowadays, machine learning models have been applied to environmental modeling with some success (Grivas and Chaloulakou, 2006; Navares and Aznarte, 2020). In this work, however, we advocate for a hybrid "culture", in which stochastic data models are combined with algorithmic ones in a way which permits harvesting the benefits of both approaches while reducing their disadvantages, as we will show.

Due to the increase in the available computing power and the advances in the field of neural network-based models, it is nowadays

[*] Corresponding author.
E-mail address: jlaznarte@dia.uned.es (J.L. Aznarte).

https://doi.org/10.1016/j.envsoft.2021.105084
Accepted 17 May 2021
Available online 28 May 2021

common to find real applications in which algorithmic modeling is put into practice to predict air quality. For example, in Nebenzal and Fishbain (2018) a system is deployed in which pollution levels based on a threshold are used to study transitions among states, which makes possible to estimate high pollution episodes in the long term. In Thatcher and Hurley (2010), authors have combined the TAPM and CCAM atmospheric models to form a customizable, local-scale meteorological and air pollution forecasting system, showing that using macroscopic models in the local scale can provide positive points in the prediction. Maciąg;g et al. (2019) is an example of an ensemble model with a neural network and an ARIMA, in a similar vein to what will be our proposal, applied successfully to a real urban environment in the city of London.

For this kind of real applications, it is useful to produce, instead of a forecast of the expected value of the magnitude under study, an estimation of the full future distribution, which in turn allows for decision making based on the probability of the surpassing of certain thresholds. This idea, which will ultimately be the main goal of the system described below, is common in other fields and was introduced to air pollution forecasting by Aznarte (2017).

The integration of meteorological information and human activities have been addressed by multiple studies. Some relevant variables, as temperature, precipitation, or wind speed have shown to be good indicators of pollution levels (Kalisa et al., 2018; Ouyang et al., 2015; Kim et al., 2015b). Also, the physical-chemical mechanisms governing the relationship between these and air quality has been studied. In Vega García and Aznarte (2020), interpretability techniques for deep learning are used to gain insight into feature importance in a highly similar environment and methodology to ours, concluding that weather variables, in general, have a high impact when using machine learning methods for predicting pollution.

However, while these issues have been widely studied from the univariate time series perspective, the observed spatial interactions between nearby observation stations might be of importance too as air quality at different stations might be implicitly related. Spatial-based approaches usually imply assuming or learning these interdependencies based for example on closeness, but, as it has been shown (de Medrano and Aznarte, 2020), this is not necessarily the most natural and optimal way to go.

In this paper, we introduce SOCAIRE (Spanish acronim for “operational forecast system for air quality”), the new official air quality monitoring system for the city of Madrid. This tool, in operational use nowadays, makes use of both external and internal variables related to air quality in order to forecast pollutant concentrations. It is a complex modular mathematical system composed of an ensemble of data manipulation techniques and models that let us exploit different knowledge in each module: from data cleaning and imputation, through handling spatial and meteorological non-linear features, to integrating human behavior and its patterns. By correctly treating all this information, it is possible to avoid redundancies and to achieve very high performance. As one of the biggest and most populated cities in Europe, Madrid is a perfect setting for developing and testing these kind of systems.

The rest of the paper is organized as follows: in Section 2 the problem is stated and Madrid’s air quality protocol for NO_2 is described, while Section 3 presents an analysis and explanation of the different data sources and the data wrangling process. Section 4 presents the proposed approach for air quality forecasting. Then, in Section 5 we introduce the Bayesian probabilistic framework that let us accommodate SOCAIRE to the NO_2 protocol. Section 6 shows the evaluation of the proposed architecture after appropriate experimentation and its comparison with other methodologies. Finally, in Section 7 we point out future research directions and conclusions.

2. Problem statement

2.1. Study area and general information

Through this work, we look for a system which is able to predict up to 48 h of four of the main existing pollutants: Nitrogen dioxide (NO_2), ozone (O_3), and particulate matter PM10 and PM2.5, where 10 and 2.5 denote the maximum diameters (in micrometers) of the particles. This estimation needs to be done in the 24 stations that compose the pollution measurement network, each one with different pollutants. Since one of the main objectives of the system is to anticipate the activation of mobility restrictions in face of high pollution episodes, we forecast the main quantiles of the distribution, so it is easier to make decisions based on pollution level probabilities. Thanks to its Bayesian estimation of compound events, SOCAIRE becomes an ideal tool to foresee the scenarios of Madrid’s NO_2 protocol, which will be explained later in this section.

SOCAIRE operates daily on a 48-h basis: it produces forecasts from 10:00 of the present day to 09:00 two days later. In the spatial dimension, the measurement stations of the city council are used as reference points. Specifically, there are 24 stations distributed throughout the city with sensors capable of recording different pollutants. Fig. 1a shows graphically the location of all the stations. At the same time, the city considers 5 different areas in the city that are related to the activation of the NO_2 protocol. These areas are shown in Fig. 1b. Table 1 shows the correspondence between the different stations and their code, their location, and the pollutants measured at each one.

2.2. The NO_2 protocol of the city of Madrid

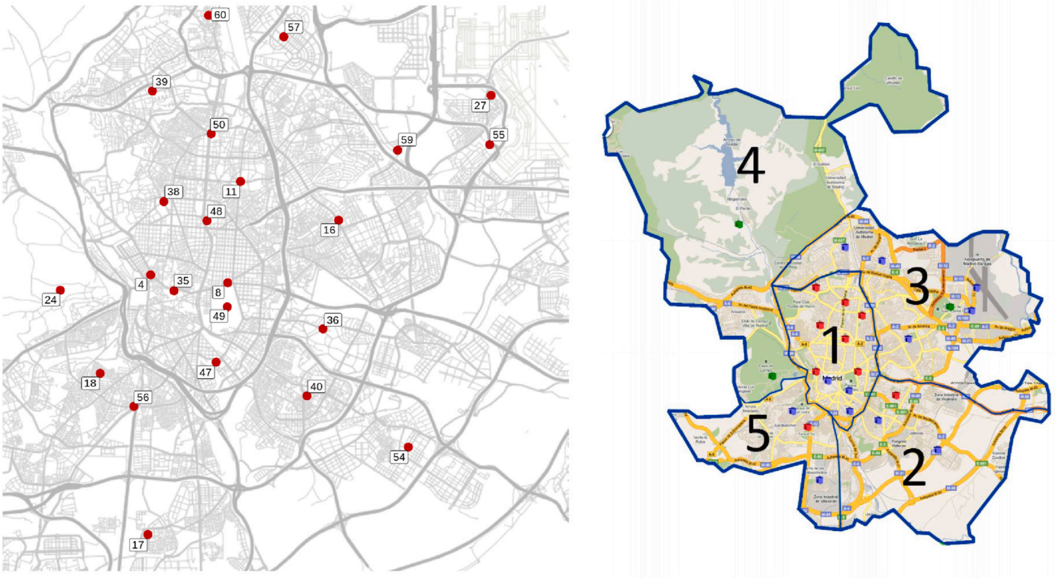
In 2018, the city council of Madrid approved an “Action Protocol for NO_2 Pollution Episodes” (Madrid-Protocol, 2018) (from this point, referred to as “the NO_2 protocol”) which defines a set of increasing alert levels, thus classifying the situations of high concentrations of NO_2 as follows:

1. PREWARNING: when any two stations in the same area simultaneously exceed 180 μgm^{-3} for two consecutive hours, or any three stations in the surveillance network simultaneously exceed the same level for three consecutive hours.
2. WARNING: when any two stations in the same area exceed 200 μgm^{-3} during two consecutive hours, or any three stations in the surveillance network exceed the same level simultaneously during three consecutive hours.
3. ALERT: when in any three stations of the same zone (or two if it is zone 4) is exceeded simultaneously, 400 μgm^{-3} during three consecutive hours.

Depending on the level and the meteorological prospect, a set of increasingly restrictive mobility limitations will be imposed city-wide by the council with the aim of mitigating and reducing the negative effects of contamination on the health and integrity of the population. Thus, the main objective is to know when and how the conditions leading to the different alert levels will be met, in order to enable the anticipation of the measures.

2.3. Framework overview

Fig. 2 presents a summary of SOCAIRE’s mathematical structure. Created to forecast and monitor pollution levels, its operation is based on the compilation of several data sources which will be described in Section 3. After a proper analysis and cleaning process, the complete database will be used through an ensemble model composed of a cascade of nested models, each one in charge of modeling different processes that alter air quality dynamics (Section 4). Finally, and thanks to the probabilistic nature of the predictions, the system is able to estimate



(a) Location of all measurement stations.

(b) Zones definition for NO_2 protocol.

Fig. 1. Location of pollutant measurement stations and definition of the 5 different areas in which the NO_2 protocol divides the city.

Table 1
Locations and availability of pollution variables for each station. A ✓reflects the presence of data in the corresponding location and for the corresponding pollutant.

Station	Code	Long.	Lat.	NO2	O3	PM10	PM2.5	Type
Pza. de España	4	−3.712	40.423	✓	–	–	–	Urban
Escuelas Aguirre	8	−3.682	40.421	✓	✓	✓	✓	Urban
Avda. Ramón y Cajal	11	−3.677	40.451	✓	–	–	–	Urban
Arturo Soria	16	−3.639	40.440	✓	✓	–	–	Urban
Villaverde	17	−3.713	40.347	✓	✓	–	–	Urban
Farolillo	18	−3.731	40.395	✓	✓	✓	–	Urban
Casa de Campo	24	−3.747	40.419	✓	✓	✓	✓	Suburban
Barajas Pueblo	27	−3.580	40.477	✓	✓	–	–	Urban
Pza. del Carmen	35	−3.703	40.419	✓	✓	–	–	Urban
Moratalaz	36	−3.645	40.408	✓	–	✓	–	Urban
Cuatro Caminos	38	−3.707	40.446	✓	–	✓	✓	Urban
Barrio del Pilar	39	−3.711	40.478	✓	✓	–	–	Urban
Vallecas	40	−3.652	40.388	✓	–	✓	–	Urban
Mendez Alvaro	47	−3.687	40.398	✓	–	✓	✓	Urban
Castellana	48	−3.690	40.439	✓	–	✓	✓	Urban
Parque del Retiro	49	−3.683	40.414	✓	✓	–	–	Urban
Plaza Castilla	50	−3.689	40.466	✓	–	✓	✓	Urban
Ensanche de Vallecas	54	−3.612	40.373	✓	✓	–	–	Urban
Urb. Embajada	55	−3.581	40.462	✓	–	✓	–	Urban
Pza. Elíptica	56	−3.719	40.385	✓	✓	✓	✓	Urban
Sanchinarro	57	−3.661	40.494	✓	–	✓	–	Urban
El Pardo	58	−3.775	40.518	✓	✓	–	–	Suburban
Juan Carlos I	59	−3.616	40.461	✓	✓	–	–	Suburban
Tres Olivos	60	−3.689	40.501	✓	✓	✓	–	Urban

probabilities from compound events using a Bayesian approach explained in Section 5 that is adapted to the aforementioned NO_2 protocol.

3. Data analysis and wrangling

As stated above, in order to aim for the highest performance, SOCAIRE makes use of all the available information related to the problem. Thus, before introducing the actual modeling, it is important to present and analyze the set of available data sources. Concretely, as anticipated, SOCAIRE uses the data of the concentrations of the different pollutants in the different stations in Madrid as dependent variables (output) and, as independent variables (inputs), past pollutant concentrations, numerical pollution predictions coming from the European CAMS model, numerical weather predictions served by AEMet, 2021 and anthropogenic information encoding different events such as holidays and school calendar. The data used along this paper corresponds to the period July 2016–October 2020, both included. The following subsections will detail the origin, peculiarities, and processing of these data.

3.1. Pollutants

The temporal behavior of each pollutant series is shown in Fig. 3. The daily cycle of all pollutants is dominated in one way or another by the

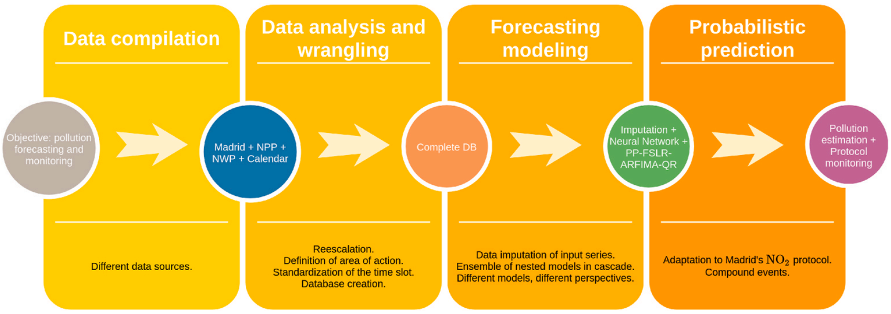

Data compilation
Data analysis and wrangling
Forecasting modeling
Probabilistic prediction
Objective: pollution forecasting and monitoring
Madrid + NPP + NWP + Calendar
Complete DB
Imputation + Neural Network + PP-FSLR-ARFIMA-QR
Pollution estimation + Protocol monitoring
Different data sources.
Reescalation.
Definition of area of action.
Standardization of the time slot.
Database creation.
Data imputation of input series.
Ensemble of nested models in cascade.
Different models, different perspectives.
Adaptation to Madrid's NO_2 protocol.
Compound events.

Fig. 2. The mathematical components of SOCAIRE. List of abbreviations: Numerical pollution predictions (NPP), numerical weather predictions (NWP), database (DB), PP-FSLR-ARFIMA-QR (Pseudo Periodic - fixed sign linear regression - ARFIMA - quantile regression).

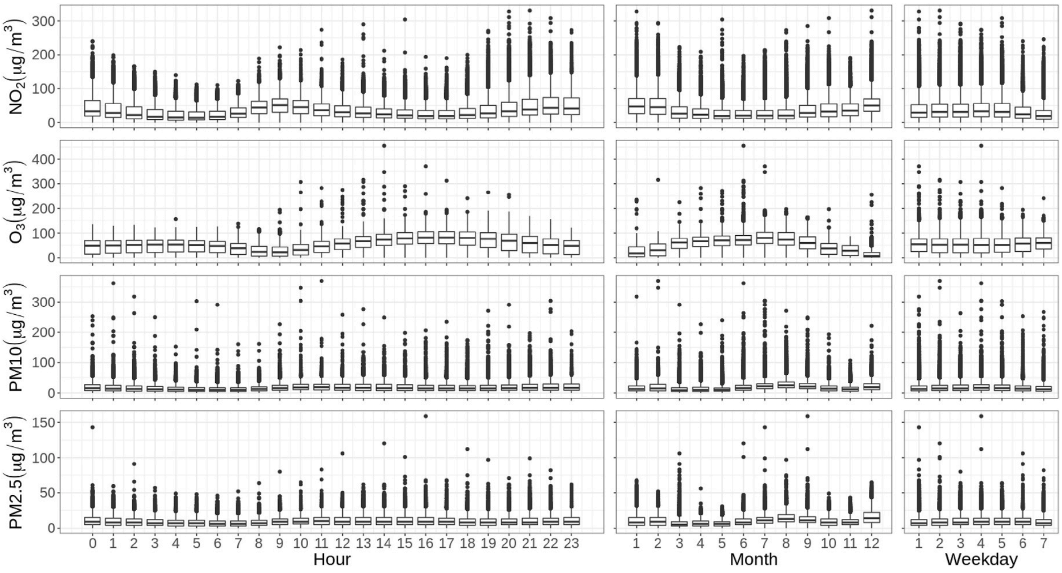


Fig. 3. Hourly, daily, and monthly temporal distribution of the four target pollutants.

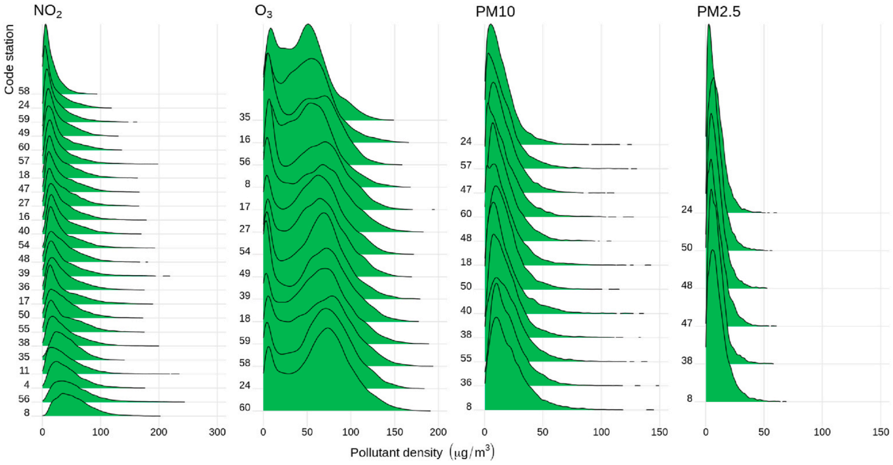


Fig. 4. Distribution of the series by pollutant and station.

peak hours of road traffic. Except for ozone, the other three pollutants to be analyzed have their daily peaks after peak traffic hours. The NO_2 has the most intense traffic-sensitive cycle, followed by the 10 and 2.5 microparticles, which show a delay of about an hour with respect to the NO_2. O_3 presents a daily cycle that is practically inverted with respect to the rest.

Everything said for the daily cycle applies to the weekly cycle, with weekend being days with lower levels of traffic. It can be assumed that holidays and long weekends will behave as public holidays, so the forecast model would have to take this into account. As expected, the daily cycle is not independent of the weekly one, but each day of the week has its own cycle, especially different on weekends from working days.

In the annual cycles, a greater variety of behaviors can be observed. All pollutants, especially ozone, rebound in summer except NO_2 which has the opposite behavior in this case.

Respect to the spatial dimension, Fig. 4 represents the empirical distributions for each pollutant. It can be seen that all stations report a similar behavior, without clear relation patterns between closeness and distribution. This fact will be of interest later when taking into account these spatial relationships in the modeling process.

As the distributions show a clear asymmetry, logarithmic transformations are used. Pollutant data is publicly available at the *Open data portal of Madrid* Madrid-Council.

3.2. Numerical weather predictions (NWP)

As mentioned in Section 1, meteorology has shown to be especially important for air quality. Hence, having weather forecasts for the period in which the air quality forecasting is being made is expected to positively impact the precision of the forecasts. In this work, we use NWP from the Integrated Forecasting System (IFS) of the ECMWF (Blanchonnet, 2015), for the following set of variables:

- **Boundary layer height** (in meters): This parameter is the depth of air next to the Earth's surface which is most affected by the resistance to the transfer of momentum, heat or moisture across the surface. The boundary layer height can be as low as a few tens of meters, such as in cooling air at night, or as high as several kilometers over the desert in the middle of a hot sunny day. When the boundary layer height is low, higher concentrations of pollutants (emitted from the Earth's surface) are found.
- **Surface pressure** (in Pa): This parameter is the pressure (force per unit area) of the atmosphere on the surface of land, sea, and in-land water. It is a measure of the weight of all the air in a column vertically above the area of the Earth's surface represented at a fixed point. Air pollution is especially prominent where high pressure dominates. Subsiding motions within an anticyclone suppress air trying to rise off the surface. Adiabatic warming of subsiding air creates a subsidence inversion which acts as a cap to upwardly moving air. Pollution problems dissipate when a low pressure system replaces a retreating anticyclone.
- **Temperature** (in K): This parameter is the temperature of air at 2 m above the surface of land, sea, or in-land waters. Generally, higher temperatures and hotwaves are directly related to episodes of higher pollution levels.
- **Precipitation** (in mm): This parameter is the accumulated liquid and frozen water, including rain and snow, that falls to the Earth's surface. It is the sum of large-scale precipitation (that precipitation which is generated by large-scale weather patterns, such as troughs and cold fronts) and convective precipitation (generated by convection which occurs when air at lower levels in the atmosphere is warmer and less dense than the air above, so it rises). Precipitation parameters do not include fog, dew, or the precipitation that evaporates in the atmosphere before it lands at the surface of the Earth. Air pollution is typically negatively correlated to the quantity of rainfall, existing a so called washing effect of precipitation.
- **U wind component** (in ms^{-1}): This parameter is the eastward component of the 10 m wind. It is the horizontal speed of air moving towards the east, at a height of 10 m above the surface of the Earth. Pollutants tend to concentrate in calm conditions, when wind speeds are not more than about 3 ms^{-1}. Speeds of 4 ms^{-1} or more favour dispersal of pollutants, which, literally, clears the air.
- **V wind component** (in ms^{-1}): This parameter is the northward component of the 10 m wind. It is the vertical speed of air moving towards the north, at a height of 10 m above the surface of the Earth. Again, wind is highly related to pollution dissemination.

NWP are interpolated to the location of each station of the air quality monitoring network. As pointed out previously, these forecasts are provided by AEMet in an hourly basis. The spatial resolution of these forecasts is $0.05 \times 0.05^\circ$ in a regular grid, while the temporal resolution is hourly, with up to 56 horizons.

3.3. Numerical pollution predictions (NPP)

CAMS, C provides a four day-horizon hourly pollution forecast which covers all Europe on a synoptic scale. The model takes into account global and regional numerical weather predictions from the ECMWF (Marécal et al., 2015), as well as other types of forecasts about the production of certain chemicals of natural and human origin from models such as C-IFS Forecasts or CAMS 81.

All these models always refer to a geodesic grid of between 10 and 20 km on each side, so it is not very sensible to use them to directly forecast the concentrations with the resolution required inside a city, which might well be below 1 km.

3.4. Anthropogenic features

As we saw in Fig. 3, depending on the human activity the temporal patterns of the series are different. Similar to weekends and months, public holidays and other designated days, as well as the school calendar, have a significant influence on road traffic, giving rise to a very different daily cycle. In special dates, we usually find a lower intensity in the center but a punctual growth in other places, particularly on the main access roads to the city related to holiday departures and returns.

Also, each type of calendar effect has different effects on each hour of the day. In addition, some of them can fall on Saturday or even on Sunday, in the case of Christmas Eve and New Year's Eve, and it is clear that the effect cannot be the same as when it falls during the week, so all these issues must be taken into consideration.

In our particular case, we will take into account the following aspects:

- **Public holidays**: Public holidays, long weekends, and special days, such as Christmas Eve and New Year's Eve, are characterized by significantly less road traffic than a normal working day (apart from other departure and return operations that may occur on some of these days and which will be taken into account later).

It has been observed that public holidays have different effects, both in terms of level and intraday evolution, depending on their location within the year, probably due to climate reasons, hours of light, and living patterns.

- **Holiday departures and returns**: Extraordinary periods such as bank holidays, long weekends, or even weekends cause a temporary exodus of citizens with large accumulations of vehicles in the so-called departure and return operations.

Departure operations can take place during the evening of the eve of the first non-working day or during the morning of that day, while return operations occur mostly during the evening of the last holiday, sometimes reaching the early morning of the next working day.

As with other variables, the effect varies with the hours within a relatively soft form.

- **School Calendar:** in Spain, school calendar and schedule is highly related to usual hourly, weekly, and monthly patterns and so, it can model with high precision the daily living. The school day can be complete or normal, average (pre- and post-holidays) or non-existent, either in isolation or for summer, winter nor spring holidays. Each type of day other than the normal one is introduced as an effect with a different intraday cycle between 07:00 and 08:00.

By combining all these variables, we ensure that the information relating to human mobility in the city is covered, both for normal situations and for special events. These exogenous variables are defined for each station, as not all parts of the city have the same dynamics.

3.5. Data wrangling

When working with such diverse data sources, is usual to deal with very heterogeneous formats and criteria, which implies that pre-processing and cleaning steps are of utmost importance. Some of the most important ones for this project are listed in this section.

Firstly, some sources use UTC time and others use Madrid's local time. In addition, the processes that transfer data between different programming environments (R, TOL, and Python) also have to take into account that each of these systems work differently with respect to winter and summer daily savings time changes.

Secondly, both NWP and NPP distribute their forecasts in a different geodesic grid, which in turn does not coincide with the coordinates of the pollution monitoring stations. At first, an attempt of interpolation was made by using the three closest grid points to each station as drivers, but it soon became apparent that this was an excessive complication with very little added value, as the forecasts were highly correlated. Therefore, in the final version, only the nearest reticular point to each station is used.

Thirdly, weather predictions are not always in the most appropriate metric, so it is necessary to create derived variables that serve better as drivers of the models. To begin with, there are variables that change scale throughout history and it is necessary to unify the criterion for obtaining uniform series in time. Then, there are other variables that are interesting to modify conceptually, for example, instead of the east-west and north-south coordinates of wind speed, it is much better to use scalar speed, which is the fundamental factor of diffusion, and direction, which is less important. Finally, it is known that meteorological factors not only have an instantaneous effect, but also a delayed effect that can be exercised up to a few hours later. For this reason, some variables delayed up to 4 h have been created and integrated with the rest of features.

Finally, since we are dealing with a cascade-like ensemble of models, in which the output of one is the input of another which may require a substantially different structure, each level of modeling requires a series of steps to prepare the data to be as expected in the next phase.

Let us note that the most laborious part of the data pre-processing has been the imputation of missing values. However, given the importance of this part, it has been decided to include imputation of data as part of the modeling strategy and is explained later in section 4.1.

4. Modeling strategy

The concentration of a given pollutant in the air depends on at least two conceptually distinct groups of factors:

- **Emission factors**: generally these are of a social order, such as road traffic or heating, which are predictable to some extent, although there are also totally unpredictable events such as fires, and others that could be anticipated to some extent such as strikes or sporting events with a multitudinous following.
- **Dispersion factors**: basically these are consequences of the weather conditions on which there are quite precise forecasts on the horizon of 2 or 3 days ahead.

Note that a certain factor, such as rain, can work in both directions at the same time: on the one hand it can cause an increase in traffic on a normal working day, which increases pollution, but on the other hand it disperses, especially the particles as they are carried to the ground, which decreases pollution. It is even possible that the effect is different depending on the day and time. Following the example of the rain that normally increases the traffic in a working day, it can on the contrary contract the traffic in an exit operation, when it will discourage people to leave the city.

This causal complexity, added to the high degree of interaction between factors, makes the phenomenon highly unstable and therefore very difficult to predict using any individual methodology. For this reason, an ensemble model composed of a cascade of nested models has been designed, such that the output of each is used in the next to get the most out of each:

- **Imputation techniques**: Although this task is usually framed as part of the data wrangling process, in this project it involves the development of models of some complexity, due to the fact that the omitted elements are presented with a certain frequency and not always in a sporadic way, but covering periods of time that can even be of several weeks. These techniques are detailed in Section 4.1.
- **NNED model**: a special flavor of convolutional neural networks called neural net encoder-decoder, which, using as inputs the outputs of the imputation models, allows to jointly forecast the concentrations of a pollutant in all the stations at the same time. It takes into account the NWP and NPP, as well as the recent past of all stations for each input variable, including the previous pollution itself, and is capable of automatically detecting non-linearities and interactions between different features. However, it does not allow for the natural treatment of irregularities in non-cyclical anthropogenic factors related with traffic. It is described in detail in Section 4.2.
- **PP-FSLR-ARFIMA-QR model**: This is a chain of models by itself developed specifically to deal with anthropogenic factors in a Bayesian way. It will be explained in detail in Section 4.3.

4.1. Imputation techniques

In the different data sources, it is relatively frequent to find missing data that can cause problems in the modeling process. For this reason, it is necessary to devise a sensible way to fill in these missing values, replacing them with approximate or expected values by a series of auxiliary models. When there are only very sporadic omissions of short duration, it might be sufficient to apply some kind of approximation by interpolation, but there might be up to consecutive weeks of data omitted in several or all variables from one or more sources at the same time. Thus, in order to develop a robust operational system, able to function even in the presence of missing data, more complex and specialized techniques are required.

4.1.1. Trigonometric interpolation

First, a trigonometric interpolation is used as a univariate method to generate sensible values for those series with clear cyclical components, such as temperature. In our case, these series present very few omissions, so we consider this technique to be sufficient. Since the data are arranged in a regular grid, this can be done by the discrete Fourier transform.

4.1.2. Multiple imputation using additive regression, bootstrapping, and predictive mean matching (HMISC)

Multiple imputation using additive regression, bootstrapping, and predictive mean matching consists of drawing a sample with replacement from the real series where the target variable is observed (i.e. not missing); fitting a flexible additive model to predict this target variable while finding the optimum transformation of it; using this fitted model to estimate the target variable in all of the original series; and finally, imputing missing values of the target with the observed value whose predicted transformed value is closest to the predicted transformed value of the missing value. This methodology is implemented in the R package *HMISC* (Jr, 2020). As the meteorological variables have already been imputed with the previous method (which will be used as input here), it is only applied to the NPP and the pollutant concentrations themselves. This method is actually used for safety in case the next one (X-ARIMA) fails. As several parts of the framework can not handle missing data, this step is required in order to assure proper functioning.

4.1.3. X-ARIMA

Once the previous two standard imputation methods are applied, it is turn for a univariate dynamic causal imputation method. It analyses how both the present and the past of a group of variables, including the target variable itself, act on the future of this target variable. These models are quite complex and, to improve the imputation, they are applied in two successive phases: in the first one, the NPP are imputed as a function of the NWP; in the second one, the pollution observations are imputed as a function of the NWP and the NPP.

Mathematically speaking, we have that, being Y_t the time series of concentration of the pollutant in question and $X_{t,k}$ the linearized inputs from the explanatory terms described above, the general formula of the Box-Jenkins' X-ARIMA models (Box et al., 1976) used is as follows (where, as usual, B is the backward operator):

$$\Delta(B)\varphi(B)\left(Y_t - \sum_{k=1}^{K} X_{t,k}\alpha_k\right) = \theta(B)\varepsilon_t. \tag{1}$$

The summation $\sum_{k=1}^{K} X_{t,k}\alpha_k$ will be called the filter of exogenous effects while the equations in differences expressed by the delay polynomials will be called endogenous factors or the ARMA part of the model.

The difference between the output and the linear filter is called ARIMA noise (z_t):

$$z_t = Y_t - \sum_{k=1}^{K} X_{t,k}\alpha_k \tag{2}$$

The previously defined backward operator delays the time indicator of some element. Mathematically speaking:

$$B^k z_t = z_{t-k} \tag{3}$$

In order to illustrate the backward notation, we may show its behavior for some simple cases. Let us suppose that the process under study presents a regular difference $\Delta(B) = 1 - B$, i.e., the difference between each pair of consecutive data is stationary.

$$\Delta(B)z_t = (1-B)z_t = z_t - z_{t-1} \tag{4}$$

Regarding to autoregressive polynomials (AR) $\varphi(B)$ and moving average (MA) $\theta(B)$, they behave similarly. Suppose we have an AR of first grade $\varphi(B) = 1 - \varphi_1 B$ and a MA of second grade $\theta(B) = 1 - \theta_1 B - \theta_2 B$. Thus, equation (1) writes as follows:

$$(1-B)(1-\varphi_1 B)z_t = (1 - \theta_1 B - \theta_2 B)\varepsilon_t \tag{5}$$

$$\left(1 - (1+\varphi_1)B + \varphi_1 B^2\right)z_t = \varepsilon_t - \theta_1\varepsilon_{t-1} - \theta_2\varepsilon_{t-2} \tag{6}$$

$$z_t - (1+\varphi_1)z_{t-1} + \varphi_1 z_{t-2} = \varepsilon_t - \theta_1\varepsilon_{t-1} - \theta_2\varepsilon_{t-2} \tag{7}$$

Note that this model is very different from the typical ARIMA model with exogenous effects of the ARIMA-X class

$$\Delta(B)\left(\varphi(B)Y_t - \sum_{k=1}^{K} X_{t,k}\alpha_k\right) = \theta(B)\varepsilon_t, \tag{8}$$

which is easier to estimate but also is considered to be much less effective in explaining the phenomena that actually occur in real life (see Appendix A).

- **Exogenous factors**: The NWP series has only very few isolated omitted data and in principle there is no reason to think that they will occur more frequently in the future. For this reason, it is more than sufficient to use an imputation system based on the Fourier transform.

The imputation of the NPP series will take as inputs the previously imputed NWP that shows quantitative relevance when imputing NPP values. Specifically, the boundary layer height (BLH), wind speed (WS), and precipitation (TP) have been used, applying different Box-Jenkins time transfer functions (Box et al., 1976) with different damping parameters in order to collect in a more synthetic way the time delayed transfers already discussed.

For the series of pollution observations, both NWP and NPP will be used, after all of them have been already imputed.

- **Endogenous factors**: The ARIMA polynomials in this case are multi-seasonal. Among the inertial factors of the stochastic process, and besides the regular time (hourly), both the daily cycle of periodicity 24 h and the weekly cycle of periodicity 24 × 7 = 168 h are taken into account.

Obviously, there is also a pseudo annual cycle and a trend but they will be filtered by some of the explanatory drivers or exogenous factors indicated in the previous section. On the one hand, the annual cycle is not in harmony with the weekly or daily cycle, that is, its periodicity is not a whole number, and on the other hand it is enormous: $365.2425 \times 24 = 8765.82$, so it is practically intractable for the ARIMA approach in an hourly series. Even in a daily series it presents serious difficulties and consumes a lot of resources.

A complete overview of the imputation process is shown in Fig. 5.

4.2. Neural network encoder-decoder: NNED model

Given that interactions between pollution itself and other relevant features, as NWP, show a complex and highly non-linear behavior in both time and space, deep learning arises as a suitable mathematical solution. No anthropogenic interactions are modeled at this point. A step forward with respect to the usual deep learning architectures, NNED model is based on the idea of spatial agnosticism for solving spatio-temporal regression problems presented in de Medrano and Aznarte (2020). It has been shown that when the spatial granularity of the series is low and its spatial autocorrelation is close to 0, traditional convolutional neural networks (CNN) fail to extract all the information from the series as the adjacency assumption for learning shared-weights does not entirely hold. That way, it is possible to obtain better prediction performance by avoiding traditional CNNs by using a spatially agnostic version of convolution.

By spatial agnostic network, we refer to a neural network in which no spatial information is introduced and past temporal information can be handled and introduced in the calculation of each new state. In order to do so, the input sequence scheme relies upon a $C \times T \times S$ images as shown in Fig. 6, where the number of channels C represents the number of input spatio-temporal variables. Similar to the usual input scheme

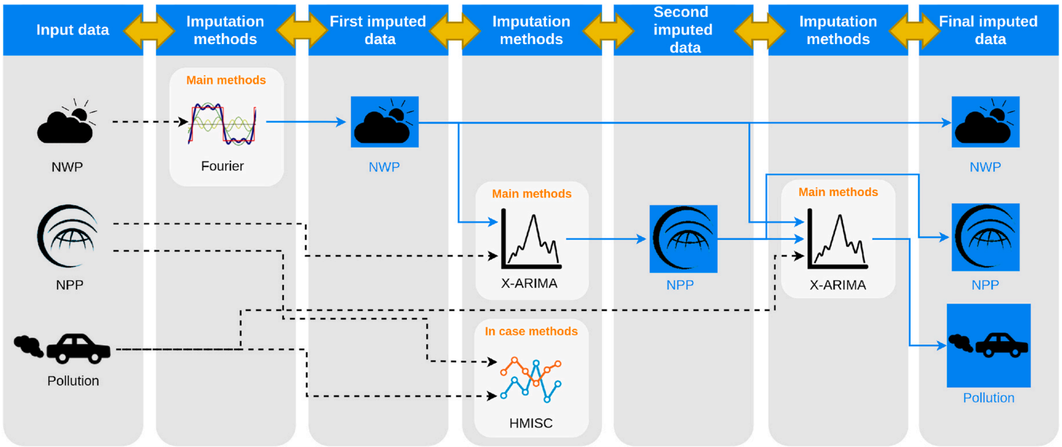


Fig. 5. Imputation scheme. All data sources likely to have missing values are part of the process. Through successive refinements, the initial data are processed through different techniques in a way that takes into account the nature and particularity of each source when performing the task. Black-dashed arrows follow processes in which data has not been imputed yet, while blue-solid arrows formalize the idea that this particular set has already been imputed.

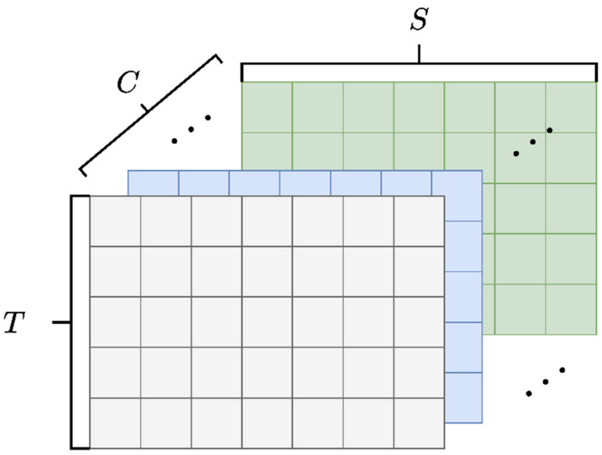


Fig. 6. Scheme for the input sequence of the NNED model. As long as all variables are spatio-temporal and have an equivalent structure for both dimensions, these sequences can be easily introduced as $C \times T \times S$ images, with variable, temporal and spatial dimension respectively.

presented in graph neural networks, this methodology let us treat both spatial and temporal dimension simultaneously. For our concrete case, the input series will be pollution, NWP and NPP for all stations during the past 48 h. The model will output pollution forecasting for all stations for the next 48 h.

NNED is composed of three different modules:

- **Encoder**: It is in charge of coding the input information of the space-time series in a space of superior dimension H. That is, it increases the expressiveness of the input by relating all the variables to each other. As we expect this model to work without spatial information, the encoder needs some modifications in its convolution scheme. The convolution itself (* operator) has the usual form for 2D images given an input x:

$$(x^{*}K)(i,j) = \sum_{m}^{k_1} \sum_{n}^{k_2} x(m,n)K(i-m,j-n) \tag{9}$$

where K is the learnable kernel. However, the kernel size is regularly used with equivalent values for its two dimensions $k_1 = k_2 = k$. In this case, not only this kernel uses different values for each component, but kernel size for spatial dimension must be equal to the number of spatial zones: $k_2 = S$. As a result, the convolution operation is made over all locations at once. The kernel size in the temporal dimension is defined as t_{past} and needs to be fixed as part of the network architecture.

The temporal dimension is dominated by a causal convolution. Generally, causal convolution ensures that the state created at time t derives only from inputs from time t to $t - t_{\text{past}}$. In other words, it shifts the filter in the right temporal direction. Thus t_{past} can be interpreted as how many lags are been considered when processing a specific timestep. Given that previous temporal states are taken into account for each step and that parameters are shared all over the convolution, this methodology might be seen as some kind of memory mechanism by itself. Unlike memory-based RNN (like LSTM and GRU) where the memory mechanism is learned via the hidden state, in this case t_{past} acts as a variable that lets us take some control over this property.

In order to ensure that each input timestep has a corresponding new state when convolving, a padding of $P = t_{\text{past}} - 1$ at the top of the image is required. To guarantee temporal integrity, this padding must be done only at the top. By using convolution in this form, once the kernel has moved over the entire input image $T \times S$, the output image will be $T \times 1$. Now, if we repeat this operation $H \times S$ times, we will create a new hidden state with H channels and an output image with $H \times T \times S$ dimensions.

Thus, we have coded input information relating all variables among them without exploiting *prior* spatial information based on adjacency.

- **Decoder**: Its function is to decode the information contained in the hidden space of high dimensionality. To do this, it learns how to merge the H hidden states present for each input and location timestep into a single value. Because this information is expected to be similar throughout the image, a kernel of size $k_1 = k_2 = 1$ is used. Thus, it changes from an image $H \times T \times S$ to, again, a $T \times S$.
- **Multilayer perceptron**: Finally, a multilayer perceptron of input $T \times S$, and output $T' \times S$ is used, relating each element obtained by the processes of coding and decoding with each of the zones and times to be predicted. The output of this multilayer perceptron is the output reported by the NNED model.

Finally, the complete procedure for this model is described graphically in Fig. 7.

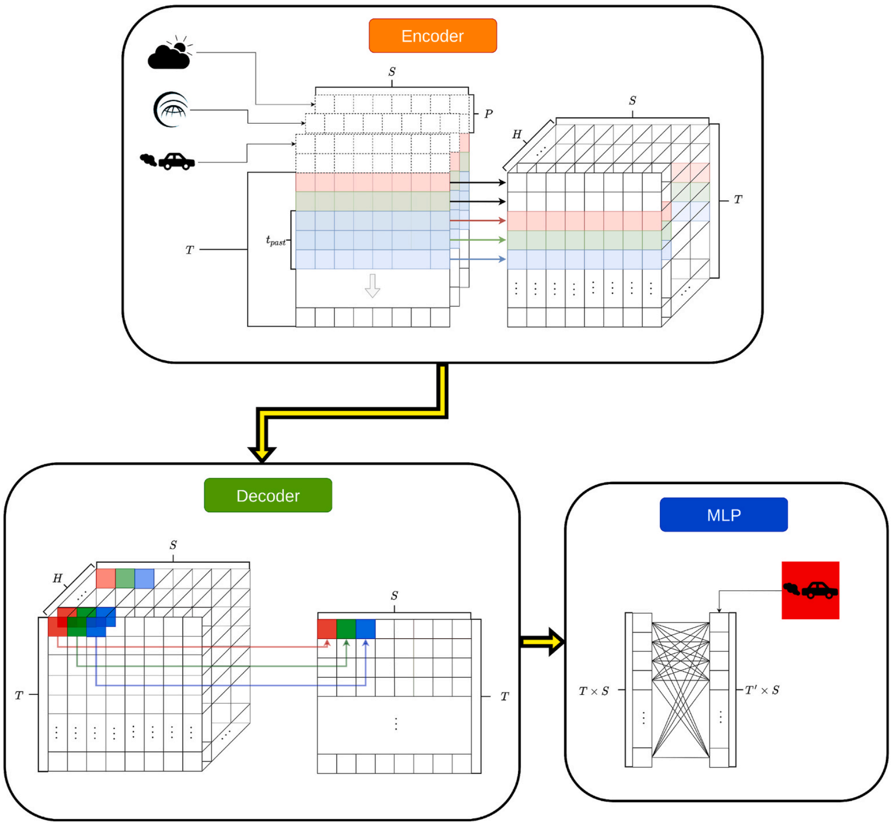


Fig. 7. NNED scheme. NNED consists of an encoder with agnostic convolutions ($k_1 = t_{\text{past}}$, $k_2 = S$), a decoder with 1 × 1 convolutions, and a dense layer that relates all input information with all output elements. In red, forecast pollutant concentration.

4.3. PP-FSLR-ARFIMA-QR model

The PP-FSLR-ARFIMA-QR model is actually a chain of models itself, which has been developed specifically to address the anthropogenic factors that in this case are of the non-cyclical calendar type. It is true that there is an underlying weekly cycle, but due to holidays and long weekends, and the interaction with the annual cycle (a long weekend in spring is not the same as in winter), it presents strong distortions that have to be dealt with *ad hoc*. Thus, this model uses the different initial data sources and knowledge learned from previous modules to exploit all this information in order to return a probabilistic prediction for the next 48 h. In this case, a different model is adjusted for each station.

4.3.1. PP: daily classification into pseudo-periodic sub-dates

Principally, the PP (Pseudo-Periodic) module is responsible for dividing the time sequence according to the type of day, depending on its position at weekends and holidays. They are called pseudo-periodic because they do not form perfect cycles like the days of the week, as the existence of holidays and long weekends disturbs their periodicity:

- **Post**: After a long weekend (usually Monday).
- **Ext**: Both the day before and the day after are working days (usually Tuesday-Thursday).
- **Prev**: Weekend or Holiday Eve (usually Friday).
- **First**: First day of a long weekend or weekend (Saturday mostly).
- **Int**: Internal to a long weekend, excluding the first and last day.
- **Last**: Last day of a holiday or weekend (Sunday as a rule).

For each one of these 6 possibilities, a time series is generated and a chain of models (described below) is developed.

4.3.2. FSLR: fixed sign linear regression

Once the type of day has been determined, we start with a linear regression whose coefficients are forced to be non-negative based on the work of Lawson and Hanson (1995). If a driver should have a negative effect, it is introduced with a change of sign. This Bayesian approach is not very common, but it is very appropriate in many occasions, since we often do not have a very detailed quantitative information about the form of the distribution of the typical prior conjugate (Fink, 1997), but we do have a very clear qualitative knowledge, for example with respect to the sign that it should take, which can be expressed as a uniform distribution in the semimark $x \geq 0$ or $x \leq 0$.

The effects considered in this regression are:

- **Instantaneous NNED forecast**: The main driver is the forecast made with the neural network model explained in Section 4.2. In the case of the **Ext** type of day it is diversified according to the day of the week which can be Tuesday, Wednesday or Thursday, as it has been observed that a certain differences exist. In the rest of sub-dating, the case of days of the week does not allow for such diversification.
- **Daily inertia (medium term)**: The average of the already known observations with 23, 24 and 25 h of delay on the one hand, and with 47, 48 and 49 on the other. By forcing the positive sign, the inertia is maintained if it is significant and positive. In other cases, the NNED algorithm itself is in charge of canceling it. It works approximately as a kind of autoregressive seasonal model of period 24 in the natural time dating, as opposed to the artificial time division subdate just described in the previous section. Concretely tree hours have been

chosen to smooth these components, considering that the daily periodicity is not completely precise in these series because of their anthropogenic component. In addition, choosing several hours avoids potential issues with the two seasonal time changes throughout the year.

- **Daily correction (medium term)**: The average of the errors made by the model itself with 23, 24, and 25 h of delay on the one hand and with 47, 48, and 49 on the other, which are also known. In this case they will be used with the opposite sign, that is, if an error is made in one direction it is corrected in the other, provided that such effect has been estimated as significant, and otherwise the NNED cancels it out. It works approximately like a kind of moving average seasonal model of period 24 in natural time dating. Concretely tree hours have been chosen to smooth these components, considering that the daily periodicity is not completely precise in these series because of their anthropogenic component. In addition, choosing several hours avoids potential issues with the two seasonal time changes throughout the year.
- **Inertia and time correction (short term)**: For the first hours of the morning of each forecast session, the observations and errors of the last hours are also available, so it is possible to build inertia and short-term correction inputs similar to the two previous ones. From midday of the same forecast day they are no longer useful. They work as a kind of regular ARMA in natural time dating.
- **Protocol Activation**: When the mobility restrictions imposed by the NO_2 protocol described in Section 2.2 are activated, the pollutant concentrations might be reduced with greater or lesser success, so that the NNED forecasts become obsolete and must be intervened in a deterministic way. They are entered with a negative sign because it would not make sense for the action to increase contamination
- **Workday indicator**: Within a long weekend, pollution is particularly reduced on the public holidays themselves, so a slight upward correction is needed for the rest of the days of the long weekend. It only affects the type of day *Int.*
- **School Calendar**: During school vacations and adaptation periods with reduced schedules at the beginning and end of the school year, there is a certain reduction in pollution that suggests a downward correction.

Concretely, this regression is estimated in logarithmic terms of both the observations and the NNED forecasts and errors, since it has been experimentally observed that the multiplicative relationship predominates over the additive.

4.3.3. Dynamic regression ARFIMA

On the errors of the previous regression, a regular dynamic model is developed (without a seasonal part) that is concerned with maintaining inertia and correcting errors produced by the anthropogenic features definition: ARFIMA. These type of models are considered as an extension of traditional ARIMA models, letting the differencing parameter to take non-integer values. By doing so, ARFIMA models are more appropriate for modeling time series with long memory (Granger and Joyeux, 1980). Through this work, the `arfima` function from the R package *forecast* is used (Hyndman and Khandakar, 2008).

4.3.4. QR probability regression

At this point, the forecasts generated represent the mathematical expectation of the output magnitudes. With this, one can aspire at most to asymptotically estimate a log-normal distribution under the laws of regression. But since the distribution will not always fit perfectly with a log-normal, it is preferable to use a method based solely on the data.

To do this, a new probabilistic Quantile Regression (QR) is estimated in order to estimate the future concentrations, with as single input the forecast of the previous FSLR+ARFIMA model, in original terms (without applying the logarithmic transformation). Quantile regression (Koenker, 2005) is an extension of linear regression used when the mean is considered insufficient to characterise the response variable. While the method of least squares estimates the conditional mean of the response variable, QR allows for the estimation of the median (q50) or, in fact, any other quantile, thus allowing for the characterisation of the full distribution of the forecasts. In our concrete case, we obtain all percentiles from 1% to 99%.

In this setting, since there is not always enough contrast surface (the data-variables ratio is low), it may happen that the estimated percentiles do not comply with the basic rules of non-negative and non-decreasing applicable to every probability distribution. Usually, it is in the extremes where there are more problems. To alleviate this inconsistency, an I-spline interpolation is applied to these percentiles to ensure that these properties are as close as possible to the estimated values.

A general schematic of the PP-FSLR-ARFIMA-QR model is presented in Fig. 8, while Fig. 9 summarizes the complete model with the data sources that govern the system.

4.4. Training procedure and operation details

From a methodological point of view, the training and parameters setting of the complete system has to be adapted to the essence of each block or model separately, since SOCAIRE presents modules of very different nature. In general terms, the data used for the training along this paper correspond to July 2016–October 2020, unless otherwise specified. The operational behavior of each of the models in relation to training and parameter estimation can be summarized as follows:

- X-ARIMA model (Section 4.1.3): The X-ARIMA parameters are estimated through bayesian methods with intellectual property reserved. These methodologies use all data available for parameter estimation, without need of hyperparameters search. This procedure is repeated each three months with all available data to that moment.
- NNED model (Section 4.2): In this case, the training follows the usual pattern of neural networks. The estimation of hyperparameters is performed by random search with data belonging to the interval January 2013–July 2016 as validation set. After this process, the network is trained with all the remaining available data using the Adam algorithm for neural parameter optimization and, once operative, the network is updated weekly by means of new optimizations that take the most recently trained network as a starting point. Every three months, a complete retraining of the network is allowed.

Some other minor details are that the network is trained using the mean squared error (MSE) as objective function. Batch size is 256, learning rate decay is set to 10^{-3}, the initial learning rate is 0.001 and both early stopping and learning rate decay are implemented in order to avoid overfitting and improve performance.

- PP-FSLR-ARFIMA-QR model (Section 4.3): As in the case of X-ARIMA, this model parameter's are estimated through bayesian methods with intellectual property reserved. Again, these methodologies can use all data available through this process. However, in this case this procedure is repeated each day with all available data as the computation require little computational power.

5. Probabilistic prediction of the alert levels

As described in Section 2.2, the activation of the NO_2 protocol depends on meeting a number of requirements, defined in three alert levels. From a probabilistic point of view, these requirements can be seen as compound events, and being able to compute the future probability for the activation of each level is of utmost importance for decision makers.

According to the NO_2 protocol, the activation of the different levels depends on what happens in several stations at the same time and in a certain number of consecutive hours. In order to compute the

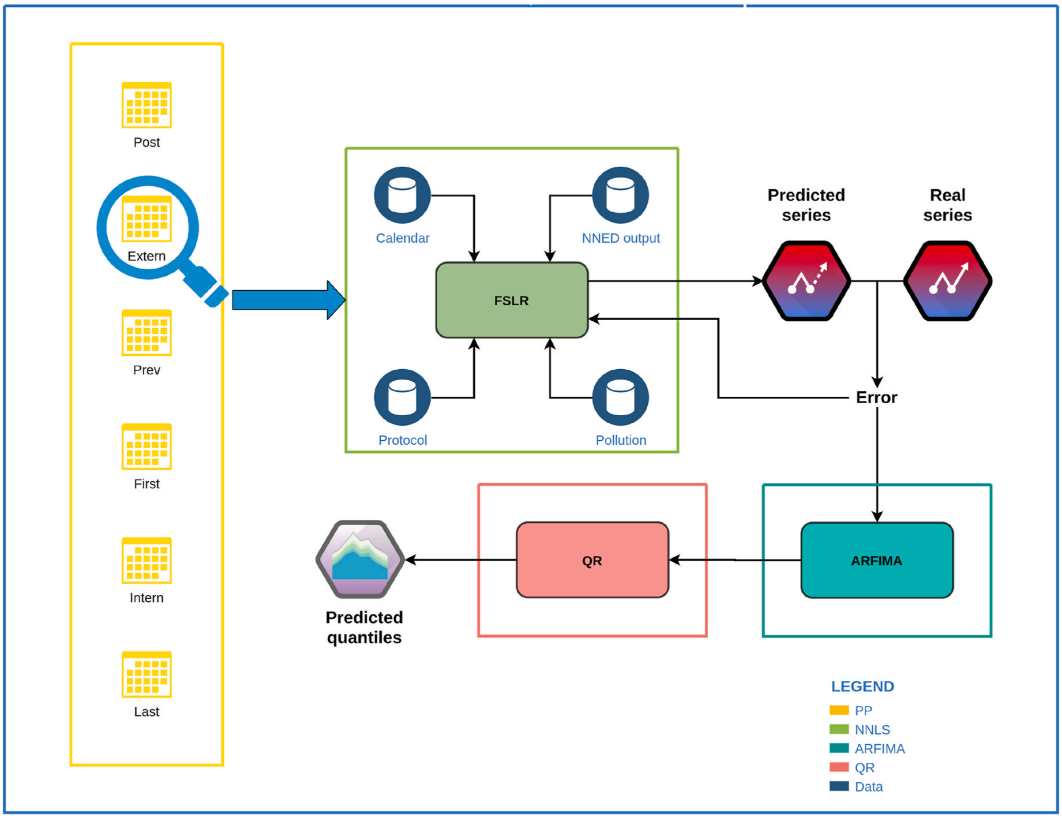


Fig. 8. PP-FSLR-ARFIMA-QR scheme. First, input data is classified depending on the type of day to be predicted (yellow). For each new series, the FSLR (green) takes as inputs different data sources (blue). Through linear relations, FSLR models different aspect of the problem, and uses its own prediction error for autoregulation. This same error is fed to the ARFIMA model (sky blue). Finally, the predicted quantiles are computed by the QR (red).

aggregated probability, the evaluation of the probability of the intersection of several events is thus needed, knowing only the marginal percentiles and the historical residues left by each of the models.

5.1. Empirical marginal distribution of the different stations

As we have shown above, the model for each station offers a probabilistic forecast condensed in a quantile vector. Specifically, the 99 integer percentiles are taken, that is, those corresponding to the probabilities $p_k = 1\%, 2\%, \ldots, 98\%, 99\%$.

In this section, we will look for a way to calculate the marginal distribution function for the forecast of each pollutant concentration from these quantiles calculated by each station's model. For this, it will be necessary to calculate the inverse of this distribution function and some statistics such as the mode, which in turn requires an analytical representation that allows us to obtain its first and second derivatives. In summary, we need a pair of easily computable, continuous, and doubly derivable functions that allow us to evaluate very efficiently and precisely approximations of the distribution function and its inverse at any point of their respective domains. The selected method is in fact an empirical change of variable that transforms the concentration into a standardized normal.

We will first take into account the fact that, by definition, the estimated quantiles are evaluations of the change in a variable that transforms the forecasts into a uniform distribution. Although this is valid for any source distribution, for reasons of numerical stability it is preferable to apply the process to the logarithms of the quantiles. Thus, if we apply the inverse of the standard normal distribution function to these log-quantiles, then the values obtained will follow that distribution by construction. Note that the calculation of the mode and deviation becomes trivial in this context.

During the approximation process, we will establish the restriction that the probability density of the concentration forecast is always unimodal, which agrees perfectly with the analyzed observations and the type of models used.

Let us think of the moment in which decision making takes place, t_0, and let us call $y_{s,t} > 0$ the real concentration not yet observed in station s at future instant $t > t_0$. The model of the s station will give us the $q_{s,t,k}$ percentiles of the forecast such that $P\left[y_{s,t} \leq q_{s,t,k}\right] = p_k$. The transformed values are thus defined as $z_{s,t} = \log\left(y_{s,t}\right)$ and the standardized normal quantile is $u_k = \Phi_{0,1}^{-1}(p_k)$, where obviously $\Phi_{0,1}$ is the normal distribution function with mean 0 and deviation 1.

Now we will interpolate the pairs $(u_k, z_{t,s})$ by means of a function $f_{s,t} : \mathbb{R} \longrightarrow \mathbb{R}$ that passes through those points

$$f_{s,t}(u_k) = z_{t,s} \tag{10}$$

and, in an analogous way, the inverse function $g_{s,t} = f_{s,t}^{-1} : \mathbb{R} \longrightarrow \mathbb{R}$ will be constructed as the interpolating function that passes through the points $(z_{t,s}, u_k)$. That is to say $g_{s,t}(z_k) = u_{t,s}$.

This allows us to construct an approximation of the concentration distribution function as follows:

$$\Phi_{0,1}\left(g_{s,t}(\log(y))\right) \simeq P\left[y_{s,t} \leq y\right] = \Psi_{s,t}(y) \tag{11}$$

And similarly we will obtain the approximation of its inverse:

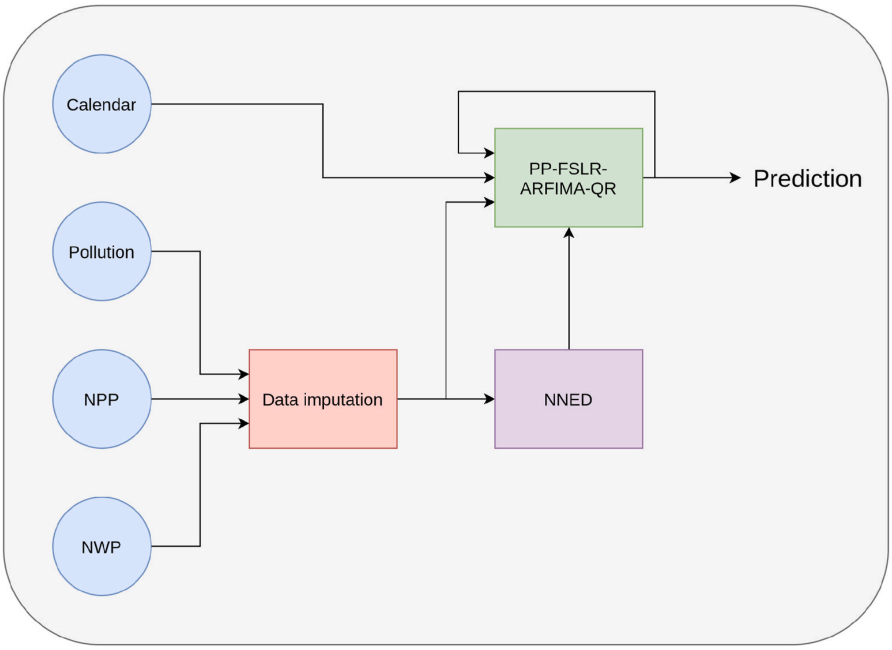


Fig. 9. Complete schematic of the ensemble of cascade nested models in SOCAIRE. In blue circles, data sources, and in squares, the mathematical components. The relationships between the different elements of the diagram are reflected by arrows.

$$\exp\left(f_{s,t}\left(\Phi_{0,1}^{-1}(p)\right)\right) \simeq \Psi_{s,t}^{-1}(p) \tag{12}$$

Although we could have directly interpolated these functions, which are the true objective, numerically speaking the interpolation with these transformations is more stable (largely because both $z_{s,t}$ and u_k are not bounded).

To avoid problems in the tails of the distribution, and taking into account that both functions are monotonous, it is highly recommended to use an interpolation method that guarantees this monotonicity. In particular, a monotonic spline interpolation has been used in this work. The monotony of the functions $f_{s,t}$ and $g_{s,t}$, together with the monotony of the logarithm and the exponential functions, guarantees that the maximum probable value of the concentration will be $\widehat{y}_{s,t} = \exp\left(f_{s,t}(0)\right)$.

Let the standardized residue of the forecast be

$$\epsilon_{s,t} = g_{s,t}\left(\log\left(y_{s,t}\right)\right) \sim N(0,1), \tag{13}$$

and note that indeed, if the probable maximum forecast is exact, i.e. if $y_{s,t} = \widehat{y}_{s,t}$, then

$$\epsilon_{s,t} = g_{s,t}\left(\log\left(\exp\left(f_{s,t}(0)\right)\right)\right) = g_{s,t}\left(f_{s,t}(0)\right) = 0 \tag{14}$$

Similarly, if the standardized residue is zero, then the forecast is exact.

5.2. *Empirical joint distribution*

Section 4 has described the models that marginally predict the concentration of each pollutant at each station for different time horizons. These models, thanks to their ARIMA structure, are able to adequately treat the internal temporal correlation of each station, that is, the autocorrelation of each of the series of pollutants of the different stations. In Fig. 10, it can be seen that the autocorrelation function (ACF) is never too big, and that when it does exceed the 2 sigma limits, so does the partial autocorrelation function (PACF). This fact suggests that these are spurious correlations or any other types of concurrent causes, not linked to time.

However, in view of Fig. 11, there is nothing that indicates that residuals from different stations will be independent of each other. Rather, they appear to correlate.

On the one hand, even if NNED models the spatio-temporal dynamics of the process, it is expected that closer stations will be more similar amongst them, giving rise to positive correlations between their residues. On the other hand, as shown in Fig. 12, the errors in each forecast horizon for a single station will also not be independent of other stations' previous horizons. In fact, this occurs mostly mutually, present errors of a station correlate with the past errors of another station and vice versa.

In the previous section we have seen how to obtain, by means of an

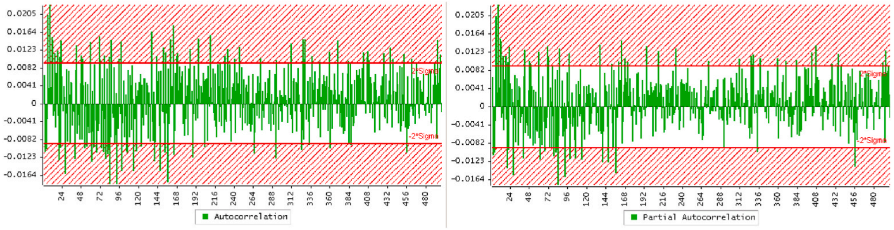


Fig. 10. ACF and PACF of residuals in station 58.

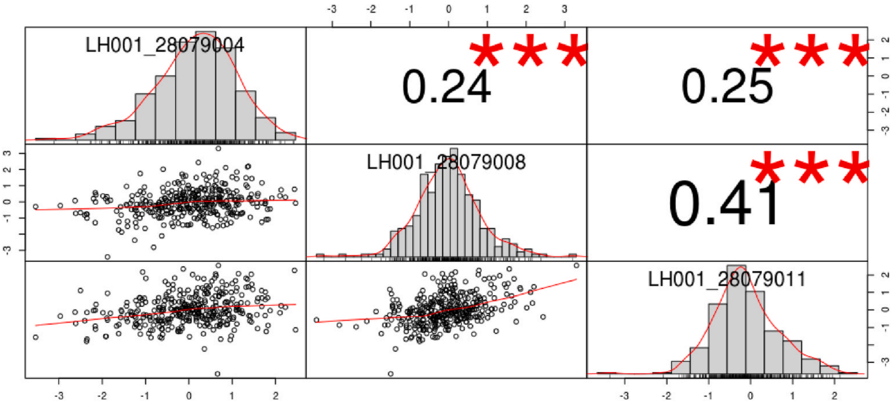


Fig. 11. Instantaneous correlations and residual histograms for stations 4, 8 and 11.

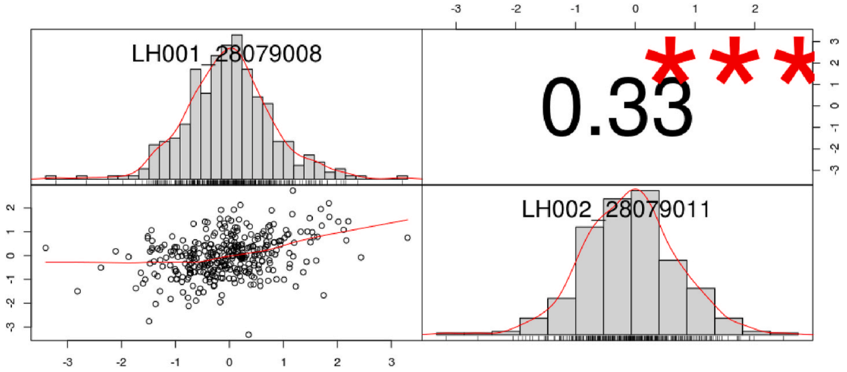


Fig. 12. Summary of the 1 h ahead cross-correlations of the residuals series for station 8 and 11.

interpolative variable change, standardized normal residues in a marginal way for each station s and for each future instant t at current time t_0. However, if the independence hypothesis is not plausible it is clear that knowing the marginal distributions does not imply knowing the joint distribution.

A family of models which are naturally capable of dealing with this situation are the X-VecARMA, a type of multivariate models (Sims, 1980) that include exogenous inputs, cointegration, and vector ARMA. They are considered very powerful for the representation of cross-correlated vector processes that might include exogenous factors eventually shared by several of them. However, they are intractable in computational terms for this setting.

Thus, we propose an empirical multi-normal copula (Nelsen, 1999) to approximate the joint distribution for every station and horizon. The aim is to obtain an estimate of such joint distribution function for all the forecasts obtained marginally, both in time and space, using the joint sample correlation matrix between each pair of stations among all the horizons and stations.

However, since there are 48 horizons and 24 stations, that gives us a square matrix of 1152 rows, and we would need at least 10 years of forecasts to obtain a meager 3 to 1 response surface, which is clearly unacceptable. For this reason, we have developed a boxed tridiagonal scheme, in which correlations are only taken into account one period ahead. With this scheme, only one year of forecasting is sufficient to obtain a reasonable estimate.

We will assume that the joint distributions of these standardized residues only depend on the station and the forecast horizons $h = t - t_0$ and $h - 1 = t - 1 - t_0$, but not on the specific moment t, since the forecasts will be made every day at the same time.

Since the marginal distributions of all the $\epsilon_{1,t}$ are normal, unbiased and with unit variance, the joint distribution of all stations,

$$\epsilon_h = \left(\epsilon_{1,t_0+h}, \ldots, \epsilon_{s,t_0+h}, \ldots, \epsilon_{S,t_0+h}\right)^T \in \mathbb{R}^S, \tag{15}$$

will be an unbiased multinormal with an unknown but obligatory unitary covariance matrix, that is, equal to the correlation matrix. In the same way we will suppose that the residuals $(\epsilon_{h-1}, \epsilon_h)$ corresponding to each pair of consecutive horizons are also distributed the same way.

By the principle of causality, for the previous horizon, ϵ_{h-1}, an independent distribution of the following ϵ_{h-1} will be postulated, since future events cannot influence the past. In this way, we can define the joint distribution of the different stations in each horizon in a recursive way:

$$\begin{gathered}
\epsilon_1 \sim \mathcal{N}(0, C_1) \\
(\epsilon_{h-1}, \epsilon_h) \sim \mathcal{N}(0, C_h), \forall h = 2, 3, \ldots, H \\
C_h = \begin{pmatrix} C_{h-1,h-1} & C_{h-1,h} \\ C_{h-1,h}^T & C_{h,h} \end{pmatrix} \in \mathbb{R}^{2S \times 2S}, \forall h = 2, 3, \ldots, H \\
C_1 = C_{1,1}, C_{h-1,h}, C_{h,h} \in \mathbb{R}^{S \times S} \\
C_{h-1,h-1,s,s} = C_{h,h,s,s} = 1 \\
C_{h,h,s,s'} = \rho_{\epsilon_{s,t_0+h}, \epsilon_{s',t_0+h}} \in (-1, 1) \\
C_{h-1,h,s,s'} = \rho_{\epsilon_{s,t_0+h-1}, \epsilon_{s',t_0+h}} \in (-1, 1)
\end{gathered} \tag{16}$$

Note that the joint distribution of all horizons would have a tridiagonal covariance matrix with partitions of order S:

$$\mathrm{C} = \begin{pmatrix} C_{1,1} & C_{1,2} & & 0 \\ C_{1,2}^{T} & \ddots & \ddots & \\ & \ddots & \ddots & C_{H-1,H} \\ 0 & & C_{H-1,H}^{T} & C_{H,H} \end{pmatrix} \tag{17}$$

If we calculate the forecasts for enough dates t_0 of the past, at the same time of the day and with the same horizons $h = 1, 2, \ldots, H$, we can obtain many samples of the residues with which we can thus estimate the matrices $C_{h-1,h}$ and $C_{h,h}$. In this way, we would obtain the distributions for each horizon conditioned on the previous horizon, using the formula known analytically for the conditional partitioned multivariate normal:

$$\left.\begin{array}{r} \epsilon_h \sim \mathcal{N}\left(\mu_h, C_h^{'}\right) \\ \mu_h = C_{h-1,h}^{T} C_{h-1,h-1}^{-1} \epsilon_{h-1} \in \mathbb{R}^{S} \\ C_h^{'} = C_{h,h} - C_{h-1,h}^{T} C_{h-1,h-1}^{-1} C_{h-1,h} \in \mathbb{R}^{S\times S} \end{array}\right\} \forall h = 2, 3, \ldots, H \tag{18}$$

These matrices can be stored for later use in future joint forecasts, along with their Cholesky and inverse decompositions:

$$\begin{array}{r} C_{h,h} = L_h L_h^{T},\ \forall h = 1, 2, 3, \ldots, H \\ C_h^{'} = L_h^{'} L_h^{'T},\ \forall h = 2, 3, \ldots, H \end{array} \tag{19}$$

First we simulate N vectors of N standardized independent residuals for the first horizon

$$\eta_{1,n} \sim \mathcal{N}(0, I), \forall n = 1, 2, \ldots, N \tag{20}$$

and pre-multiplying them by L_1 we will have the standardized residuals of all the stations for the first horizon:

$$\epsilon_{1,n} = L_1 \eta_{1,n} \sim \mathcal{N}(0, C_1) \tag{21}$$

From there, also starting from independent residuals

$$\eta_{h,n} \sim \mathcal{N}(0, I), \forall n = 1, 2, \ldots, N, \tag{22}$$

residuals of each horizon conditioned by the previous one can be simulated:

$$\epsilon_h = \mu_h + L_h^{'} \eta_{h,n}. \tag{23}$$

On the one hand, this approach solves the problem of time correlation in consecutive hours, which is what is required, and on the other hand it is simple enough to be able to generate correct estimations.

Finally, applying the transformations detailed Section 5.2 we obtain N realizations of the future forecasts of the concentrations of the different stations in each horizon:

$$y_{s,t_0+h,n} = \exp\left(f_{s,t}\left(\epsilon_{s,h,n}\right)\right) \tag{24}$$

If this simulation is repeated a sufficient number of times we can calculate any joint statistic from the forecasts of the concentrations in the different stations. In particular, for example, to calculate the probability of activation of the pre-warning level of the NO_2 protocol, defined as the probability of the concentration of NO_2 exceeding a certain threshold $\Upsilon = 180$ in at least two stations during two consecutive hours, it will simply be necessary to calculate what proportion of the simulated samples meet these criteria.

6. Operation and performance

6.1. Operation

In order to be used by decision makers in the department of the city council in charge of air quality, SOCAIRE has been integrated with a web app that allows to simply and directly view the forecasts for pollutants and the probability of reaching the levels established within the NO_2 protocol as explained in section 2.2. This section will show the site structure and its basic operation principles.

The main overview of the web tool is shown in Fig. 13. On the one hand, at the top you can choose the pollutant to display (blue buttons), the date on which you want to make a query (calendar button), and different submenus where you can see in more detail the probability that the protocol will be activated (shown tab), and both the system predictions and a summary of contrast measures. On the other hand, in the central part the information related to the submenu in which the user is at that moment is shown. In this specific case, the probability of the levels of the NO_2 protocol being activated.

The operation of the tool for monitoring the future probability of reaching the different levels of the protocol are presented in Fig. 14. After using the ensemble of nested models described in Section 4 to forecast NO_2 quantiles, the outer rings show the probability of each individual station exceeding the levels set in the NO_2 protocol (180 μg/m^3, 200 μg/m^3, and 400 μg/m^3 for prewarning, warning, and alert respectively). Once the individual probabilities are computed, it is possible to use the process explained through Section 5 to estimate probabilities of compound events. Given that the protocol is defined over areas and not for individual station levels, the intermediate ring shows the probability of exceeding the expected pollution levels for each of the 5 areas in which Madrid is partitioned in the NO_2 protocol (see Fig. 1b). Lastly, the inner ring contains the aggregated probability of the different levels of the protocol being activated in the entire city. It uses the probabilities over the five areas to estimate this final probability.

Since the set of mobility measures defined in the NO_2 protocol depends on reaching extreme levels in various stations and for a pre-set number of consecutive hours, having such an overview is especially important. However, it is also interesting to visualize the individual forecast for each station over time. The SOCAIRE website allows viewing the actual forecasts for each pollutant and each station, as shown in Fig. 15. Together with the predicted quantiles and real observed values, these plots also show the probability of exceeding each level and the levels themselves.

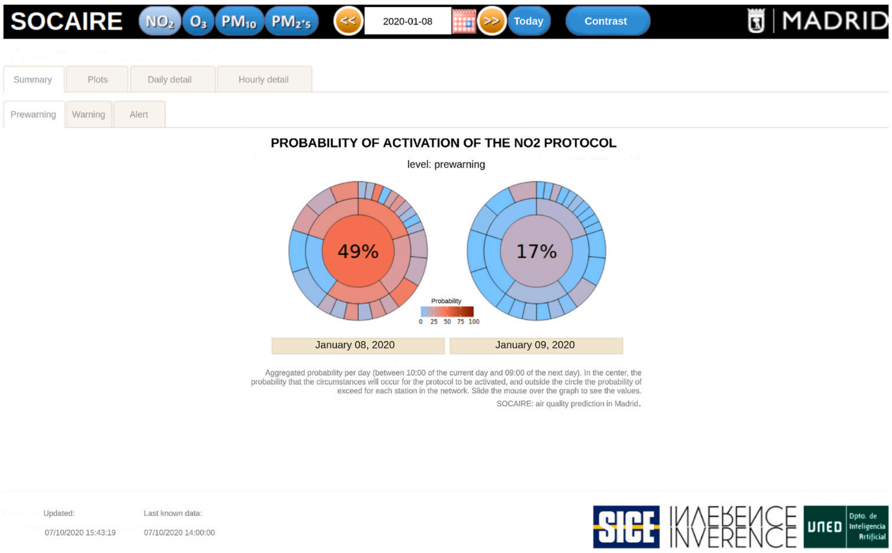
6.2. Performance analysis

Usual error metrics, as RMSE, refer to expected values, which are found in the central part of the distribution, but do not take into account any other information, and are thus particularly unfit to evaluate probabilistic forecasts. Since the most usual models produce point forecasts and not the entire distribution, these kinds of metrics are the only option. However, when dealing with the prediction of the complete distribution as in our case, other metrics have been proposed in order to summarise model performance information in a more comprehensive and realistic way. For example, CRPS is a measure of the squared difference between the forecast cumulative distribution function (CDF) and the empirical CDF of the observation (Gneiting and Katzfuss, 2014).

As we will show, in terms of performance SOCAIRE compares favorably to benchmarks. In order to get a clear and quick idea about the behavior of the model, Table 2 shows the RMSE and bias (averaged both in time and space) of the proposed methodology and compares it with four other models that, due to their characteristics, make it easier to understand the real performance of SOCAIRE: persistence, linear regression, NNED output without any linear correction, and the NPP provided by CAMS.

The persistence model is a naive model in which the forecast value is taken to be the observed value at the previous timestep. It is, thus, a good benchmark model and one can get a rough idea of how good a new model is by seeing how much improvement there is with respect to persistence. In our specific case, for contractual reasons, we use a more elaborated version of persistence which includes the daily, weekly, and annual cyclical structure of the series, and is thus a simple although powerful model.

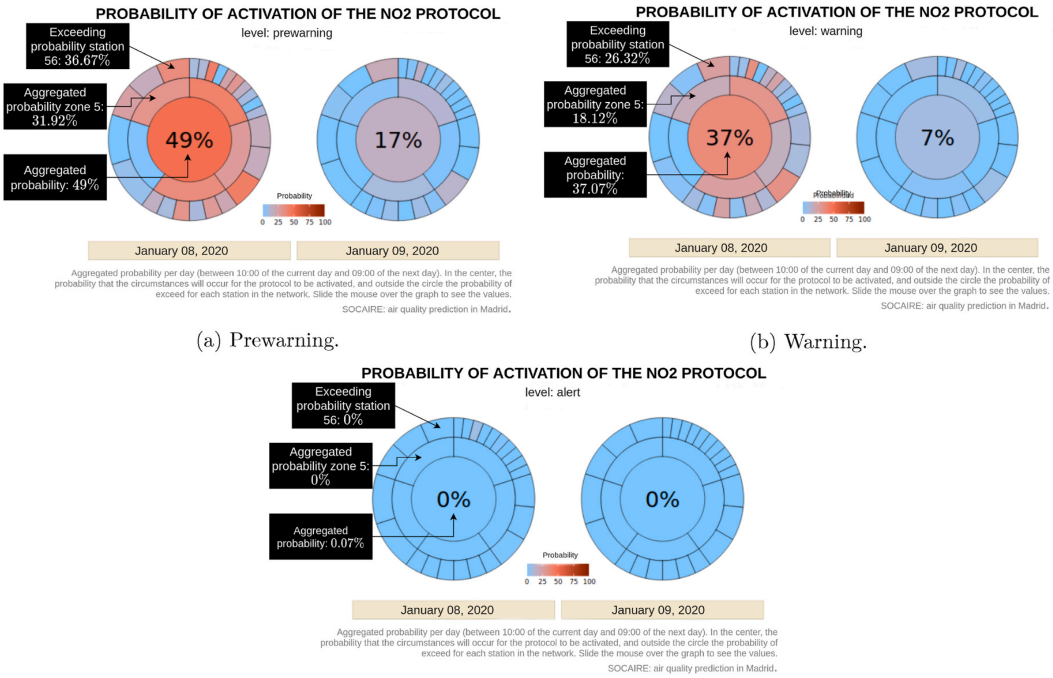
Linear regression is a well known methodology for all kind of



Aggregated probability per day (between 10:00 of the current day and 09:00 of the next day). In the center, the probability that the circumstances will occur for the protocol to be activated, and outside the circle the probability of exceed for each station in the network. Slide the mouse over the graph to see the values.

SOCAIRE: air quality prediction in Madrid.

Fig. 13. Main page of the SOCAIRE web app for controlling and monitoring pollution in the city of Madrid.



Aggregated probability per day (between 10:00 of the current day and 09:00 of the next day). In the center, the probability that the circumstances will occur for the protocol to be activated, and outside the circle the probability of exceed for each station in the network. Slide the mouse over the graph to see the values.

SOCAIRE: air quality prediction in Madrid.

(a) Prewarning.

Aggregated probability per day (between 10:00 of the current day and 09:00 of the next day). In the center, the probability that the circumstances will occur for the protocol to be activated, and outside the circle the probability of exceed for each station in the network. Slide the mouse over the graph to see the values.

SOCAIRE: air quality prediction in Madrid.

(b) Warning.

Aggregated probability per day (between 10:00 of the current day and 09:00 of the next day). In the center, the probability that the circumstances will occur for the protocol to be activated, and outside the circle the probability of exceed for each station in the network. Slide the mouse over the graph to see the values.

SOCAIRE: air quality prediction in Madrid.

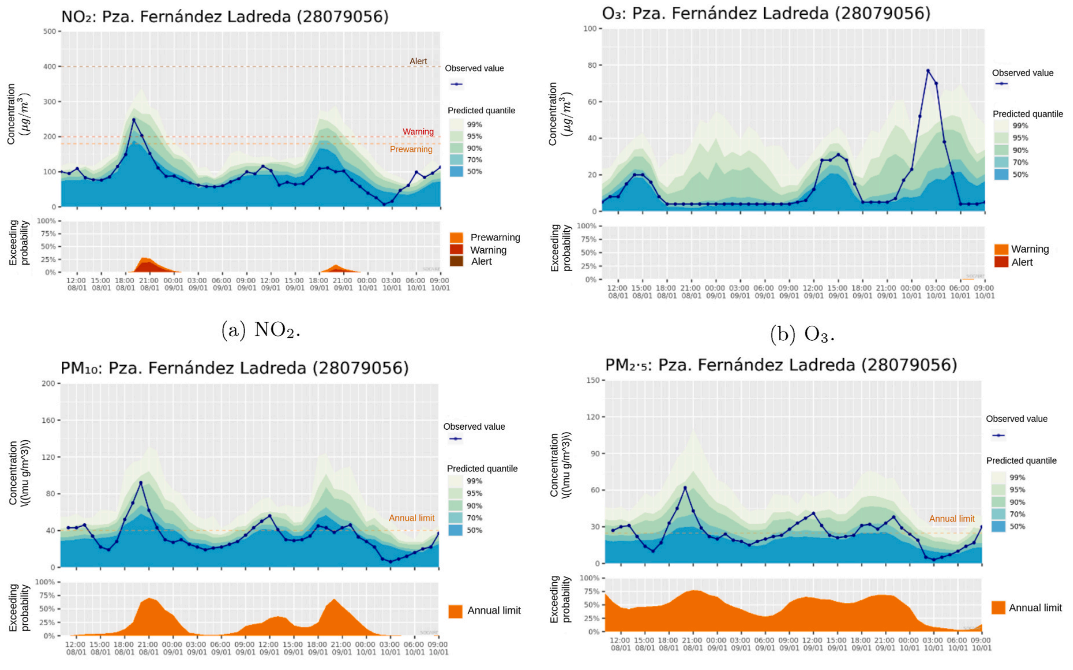
(c) Alert.

Fig. 14. Probability of activation for the three levels of the NO_2 protocol, predicted the September 29, 2020.

regression problems, characterized by its simplicity but performing reasonably well in a multitude of scenarios. In its most basic version it is limited by its only linear response, so it is a good candidate to be beaten as a sample of having a model of minimum guarantees, as was the case with persistence. In our particular case, we use a multioutput scheme: for each station we generate a model that will have as input the past timesteps and will return jointly all predicted timesteps.

Regarding the NNED model, its inclusion has a dual purpose: on the one hand, to have a clear and direct comparison with a neural architecture, on the other hand, to be able to clearly and precisely visualize the improvement that the complete modeling explained in Section 4 implies in terms of performance and the potential benefit that can be obtained from using both types of strategies.

Similarly, the NPP provided by CAMS represent another good baseline to be improved upon by any new model. Since it is based on a synoptic scale, it is expected that any model focused on a smaller and concrete terrain extension will improve its results. If this is not the case, it would make more sense to use CAMS NPP as an approximation instead



(a) NO$_2$.

(b) O$_3$.

(c) PM10.

(d) PM2.5.

Fig. 15. Forecast quantile and probabilities of exceedance for each hour in station 56 for January 08, 2020. The real observed values are represented by the blue lines and dots and, thus, it is possible to have a reference about SOCAIRE performance (which will be covered in Section 6.2).

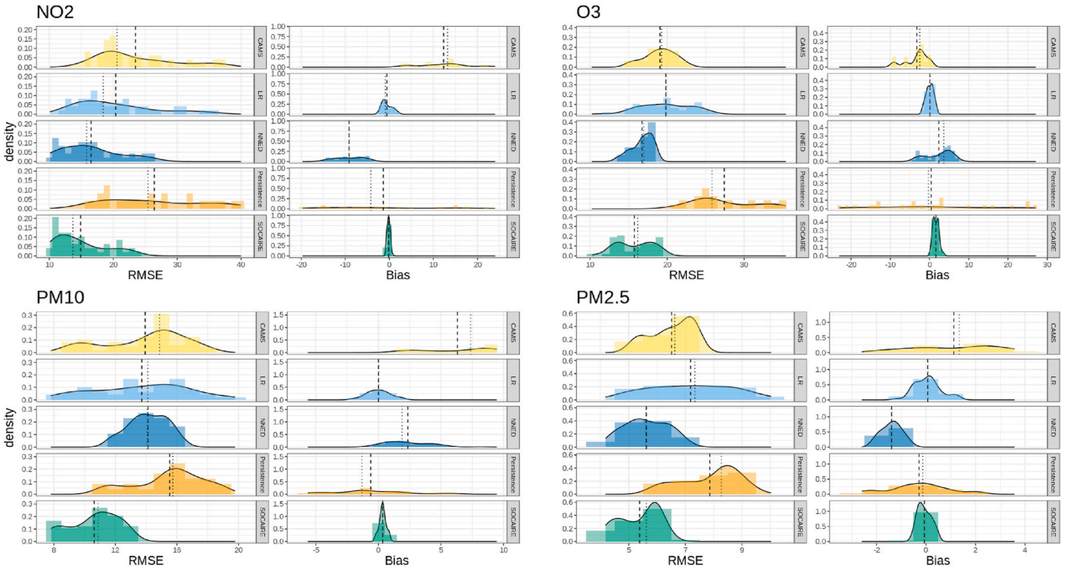


Fig. 16. Error distribution for the four pollutants in terms of RMSE and bias. Dashed vertical line represents the mean, dotted vertical line represents the median.

of the proposed new methodology.

For a more detailed view of error metrics, refer to Fig. 16. As it can be seen, SOCAIRE consistently outperforms all baselines in terms of RMSE and bias for the four pollutants. Concretely, SOCAIRE supposes an average RMSE improving of 37% with respect to CAMS, a 27% with respect to LR, a 10% with respect to NNED, and 44% with respect to persistence, reinforcing the idea that SOCAIRE shows good performance and behaves very well as a predictor. Also, SOCAIRE demonstrates to be in general terms an unbiased predictor of pollution, which emphasizes the fact that the proposed model is being able to correctly describe the aforementioned terms related to the system.

Another issue that is of special importance in our problem is the behavior of the model depending on the prediction timestep/hour. As it was shown in Fig. 3, the series are highly hour-dependent. For example, NO$_2$ presents peaks usually around 08:00–10:00 and 22:00–00:00. In the framework of air quality management and monitoring, these peaks are

Table 2
Average error for $t = 1$ to $t = 48$, calculated over all stations. For a more detailed view of error metrics distribution, see Fig. 16.

	NO_2		O_3		*PM*10		*PM*2.5	
	RMSE	Bias	RMSE	Bias	RMSE	Bias	RMSE	Bias
CAMS	23.5 ± 9.1	12.3 ± 9.8	19.1 ± 5.0	−3.2 ± 6.2	13.9 ± 3.7	6.3 ± 3.7	6.5 ± 1.3	1.1 ± 2.1
LR	20.7 ± 6.3	−0.1 ± 1.8	20.5 ± 3.9	−0.8 ± 1.1	13.4 ± 2.9	0.3 ± 0.7	7.2 ± 1.3	0.2 ± 0.5
NNED	16.5 ± 4.5	−9.2 ± 3.2	16.8 ± 1.4	2.4 ± 3.3	14.2 ± 1.3	2.4 ± 1.8	5.6 ± 0.8	−1.4 ± 0.4
Persistence	26.4 ± 9.3	−1.4 ± 3.5	27.4 ± 4.8	0.5 ± 16.0	15.5 ± 3.1	−0.6 ± 3.5	7.8 ± 1.4	−0.3 ± 1.4
SOCAIRE	14.9 ± 4.8	−0.2 ± 0.8	15.8 ± 2.8	1.6 ± 1.0	10.6 ± 2.5	0.3 ± 0.7	5.4 ± 1.0	−0.1 ± 0.6

extremely important as they represent the higher risk and, consequently, the moments when the maximum recommended and/or permitted levels are usually exceeded. Thus, and given that one of the main objectives of SOCAIRE framework is forecasting the probability of each level of the NO_2 protocol, showing a good performance in peak hours is of crucial importance.

Fig. 17 presents the RMSE error for each pollutant and for each prediction horizon averaged over all stations. From this figure, it becomes clear that SOCAIRE is especially efficient in peak hours, where the gap with baseline models is even wider.

Until now, we have covered aggregated error over all stations. As the activation of the NO_2 protocol depends on compound events of individual stations, it is important to make sure all of them behave similarly. As it was explained before, the complete model has a module which is able to relate and exploit shared spatial information (Section 4.2), but it also models each station independently based on its own characteristics (Section 4.3). By taking into account both types of information, we expect to avoid possible biases of predominance by some spatial areas over others but still be able to make use of the relations that exist among them. The CRPS for the NO_2 predictions at each station is shown in the top row of Fig. 18. It is worth noting that stations with lower CRPS errors correspond to green areas of the city of Madrid (Stations 24, 49, and 58). Scaling these CRPS values to a $\mathcal{N}(0,1)$ (bottom row of Fig. 18) let us see how all error distributions have a very similar behavior. Hence, it is possible to assure that our modeling strategy works as expected and results in an approximately unbiased prediction of the spatial component.

The evaluation of these models has been done using the data from January 2020 to October 2020, with the system already operational and therefore functioning as described in Section 4.4. Thus, the estimated errors represent realistically the errors the model is recording in its daily operation. The aggregated error metrics from all predicted timesteps over that period generate the error distributions analyzed in this section.

7. Conclusions and future work

Throughout this manuscript, we have discussed the details of SOCAIRE, the new operational system for air quality forecasting and monitoring in the city of Madrid. Based on an ensemble of statistical and neural models, SOCAIRE is built under the premise that it is possible to integrate the diverse information that correlates with air quality in order to model it. This information includes historical values of the series itself, numerical weather and pollution predictions, and anthropogenic features. Concretely, the proposed methodology tackles the prediction of the four main pollutants (NO_2, O_3, PM10, and PM2.5) for a 48-h horizon. Thanks to its probabilistic nature, the system is able to combine the predictions of the full probability distribution for compound events using a Bayesian estimation of the future distribution of the different stations over time. Thus, the system outputs are a valuable tool for managing the NO_2 protocol enforced by the city council of Madrid.

The tool presented in this paper is not only a theoretical proposal, but it has been adopted as the official application to monitor, analyze and make day-to-day decisions about air quality. The last part of this work summarizes the structure and operation of SOCAIRE's web, as well as the main highlights of the good results and performance of the system.

In the future, it would be interesting to apply a cost-effectiveness analysis focused on the NO_2 protocol activation probability. Also, we are working towards the inclusion of a traffic forecasting system, which might improve the performance of the models by enhancing the

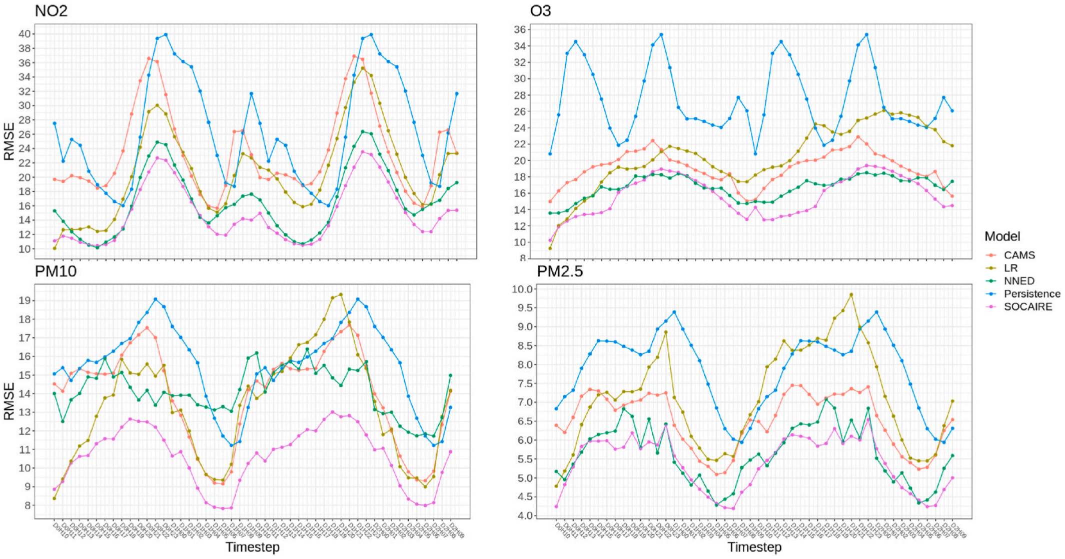


Fig. 17. RMSE error by timestep. D makes reference to the day and H to the hour (D0H10 means day 0 or present day at 10:00).

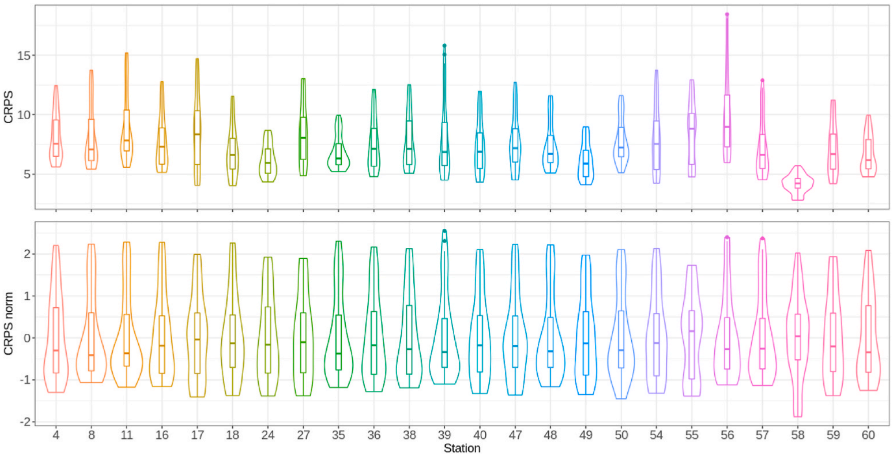


Fig. 18. Comparison of the error distribution for all stations (top). Scaling CRPS to a $\mathcal{N}(0,1)$ let us conclude that all stations have been correctly modeled in the spatial dimension (bottom).

information that anthropogenic features provide. Finally, SOCAIRE could be adapted to predict any kind of combined air quality index, and not only those ones affecting the current protocol.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank José Amador Fernández Viejo and the team at the General Directorate for Sustainability of the Municipality of Madrid, especially María de los Ángeles Cristóbal López for her continuing support and enthousiasm towards this project.

This research has been partially funded by Empresa Municipal de Transportes (EMT) of Madrid, Spain under the program Cátedra *EMT/UNED de Calidad del Aire y Movilidad Sostenible*".

A X-ARIMA vs ARIMA-X

In order to illustrate the differences between both kinds of models, let us introduce the simplest case: AR(1). Specifically, $\Delta(B) = 1$, $\varphi(B) = 1 - \varphi_1 B$, and $\theta(B) = 1$.

In this circumstances, ARIMA-X would be reduced to a linear regression:

$$Y_t = \alpha X_t + \varphi_1 Y_{t-1} + \varepsilon_t \tag{25}$$

whereas the X-ARIMA model would be bilinear, which is much more complicated to estimate than a linear regression.

$$Y_t = \alpha X_t + \varphi_1 (Y_{t-1} - \alpha X_{t-1}) + \varepsilon_t \tag{26}$$

Note that when the absolute value of φ_1 is very small there will be almost no difference between the two models but otherwise they will be very different.

Let us imagine for simplicity that there is a single exogenous driver consisting of a pulse (the blue line in Fig. 19), i.e., its value is 1 at a given instant of time and 0 the rest of the time. We have arbitrarily set the parameters $\sigma = 0.1$, $\alpha = 2$ and $\varphi_1 = 0.93$ and simulated two processes, each following one of the models: in red with a thicker line the X-ARIMA and in orange the ARIMA-X, both generated from the same series of residuals (green line). Logically both series coincide perfectly until the pulse occurs, but, while in the first one the effect of the pulse vanishes instantaneously, in the second one it lasts quite a long time because the AR root is very close to unity. If we had set $\varphi_1 = 0.70$ the effect would have lasted not 12 h but almost two days.

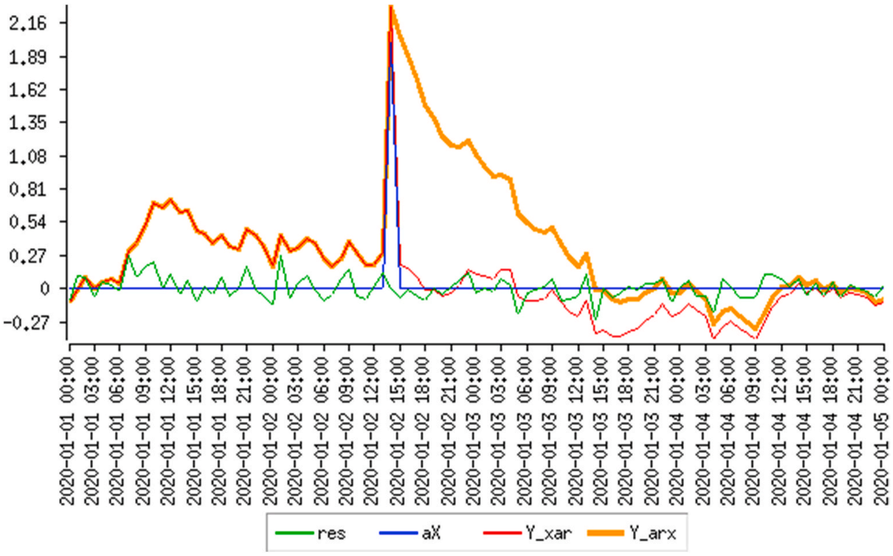


Fig. 19. Comparison between X-ARIMA and ARIMA-X models.

While usually instantaneous transfers are much more common than damped transfers like the one we have shown, even when they occur, they do not usually present exactly the same shape and damping rate as the series noise itself. Although possible, the probabilities of all transfer functions of all the inputs being coincident with each other and with the ARIMA model are scarce.

Strictly speaking, using the appropriate transfer functions the two model classes are equivalent, but X-ARIMA fits in a more natural way and without using complicated constraints.

References

AEMet, 2021. Agencia Estatal de Meteorología, Gobierno de España. URL: http://www.aemet.es/es/portada.

Aznarte, J.L., 2017. Probabilistic forecasting for extreme NO2 pollution episodes. Environ. Pollut. 229, 321–328. https://doi.org/10.1016/j.envpol.2017.05.079.

Badyda, A.J., Grellier, J., Dąbrowiecki, P., 2017. Ambient PM2.5 exposure and mortality due to lung cancer and cardiopulmonary diseases in polish cities. Adv. Exp. Med. Biol. 944, 9–17. https://doi.org/10.1007/5584_2016_55.

Bai, L., Wang, J., Ma, X., Lu, H., 2018. Air pollution forecasts: an overview. Int. J. Environ. Res. Publ. Health 15. https://doi.org/10.3390/ijerph15040780.

Blanchonnet, H., 2015. Set I - atmospheric Model high resolution 10-day forecast (HRES). https://www.ecmwf.int/en/forecasts/datasets/set-i. library Catalog: www.ecmwf.int.

Box, G.E., Jenkins, G.M., Reinsel, G.C., Ljung, G.M., 1976. Time Series Analysis: Forecasting and Control. John Wiley & Sons.

Breiman, L., 2001. Statistical modeling: the two cultures (with comments and a rejoinder by the author). Stat. Sci. 16, 199–231. https://doi.org/10.1214/ss/1009213726. https://doi.org/10.1214/ss/1009213726.

AMS, . Copernicus air quality monitoring. URL: https://atmosphere.copernicus.eu/.

European Union, P., 2008. Directive 2008/50/EC of the European Parliament and of the Council of 21 may 2008 on ambient air quality and cleaner air for Europe. Off. J. European Union.

Fink, D., 1997. A Compendium of Conjugate Priors. Environmental Statistical group, Department of Biology, Montana State University, USA.

Gneiting, T., Katzfuss, M., 2014. Probabilistic forecasting. Annual Rev. Stat. Appl. 1, 125–151. https://doi.org/10.1146/annurev-statistics-062713-085831. https://doi.org/10.1146/annurev-statistics-062713-085831. doi:10. 1146/annurev-statistics-062713-085831. _eprint.

Granger, C.W.J., Joyeux, R., 1980. An introduction to long-memory time series models and fractional differencing. J. Time Anal. 1, 15–29. https://doi.org/10.1111/j.1467-9892.1980.tb00297.x. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-9892.1980.tb00297.x.

Grivas, G., Chaloulakou, A., 2006. Artificial neural network models for prediction of PM10 hourly concentrations, in the Greater Area of Athens, Greece. Atmos. Environ. 40, 1216–1229. https://doi.org/10.1016/j.atmosenv.2005.10.036.

Hassanzadeh, S., Hosseinibalam, F., Alizadeh, R., 2009. Statistical models and time series forecasting of sulfur dioxide: a case study Tehran. Environ. Monit. Assess. 155, 149–155. https://doi.org/10.1007/s10661-008-0424-1.

Hyndman, R.J., Khandakar, Y., 2008. Automatic time series forecasting: the forecast package for R. J. Stat. Software 26, 1–22. https://www.jstatsoft.org/article/view/v027i03.

Héroux, M.E., Anderson, H.R., Atkinson, R., Brunekreef, B., Cohen, A., Forastiere, F., Hurley, F., Katsouyanni, K., Krewski, D., Krzyzanowski, M., Künzli, N., Mills, I., Querol, X., Ostro, B., Walton, H., 2015. Quantifying the health impacts of ambient air pollutants: recommendations of a WHO/Europe project. Int. J. Publ. Health 60, 619–627. https://doi.org/10.1007/s00038-015-0690-y.

Harrell Jr., Frank E, with contributions from Charles Dupont and many others, 2020. Hmisc: Harrell Miscellaneous. R package version 4.3-1. https://CRAN.R-project.org/package=Hmisc.

Kalisa, E., Fadlallah, S., Amani, M., Nahayo, L., Habiyaremye, G., 2018. Temperature and air pollution relationship during heatwaves in Birmingham, UK. Sustain. Cities Soc. 43, 111–120. https://doi.org/10.1016/j.scs.2018.08.033.

Kim, K.H., Kabir, E., Kabir, S., 2015a. A review on the human health impact of airborne particulate matter. Environ. Int. 74, 136–143. https://doi.org/10.1016/j.envint.2014.10.005.

Kim, K.H., Lee, S.B., Woo, D., Bae, G.N., 2015b. Influence of wind direction and speed on the transport of particle-bound PAHs in a roadway environment. Atmos. Pollut. Res. 6, 1024–1034. https://doi.org/10.1016/j.apr.2015.05.007.

Koenker, R., 2005. Quantile Regression. Econometric Society Monographs, Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511754098.

Kumar, U., Jain, V.K., 2010. ARIMA forecasting of ambient air pollutants (O3, NO, NO2 and CO). Stoch. Environ. Res. Risk Assess. 24, 751–760. https://doi.org/10.1007/s00477-009-0361-8.

Lawson, C.L., Hanson, R.J., 1995. Solving Least Squares Problems. SIAM.

Maciąg, P.S., Kasabov, N., Kryszkiewicz, M., Bembenik, R., 2019. Air pollution prediction with clustering-based ensemble of evolving spiking neural networks and a case study for London area. Environ. Model. Software 118, 262–280. https://doi.org/10.1016/j.envsoft.2019.04.012.

Madrid City Council, 2021. Portal de datos abiertos del Ayuntamiento de Madrid. https://datos.madrid.es/portal/site/egob.

Madrid-Protocol, 2018. Protocolo de actuación para episodios de contaminación por dióxido de Nitrógeno. https://www.madrid.es/portales/munimadrid/es/Inicio/Medidas-especiales-de-movilidad/Protocolo-de-contaminacion/Protocolo-de-actuacion-para-episodios-de-contaminacion-por-dioxido-de-nitrogeno/?vgnextfmt=default&vgnextoid=fd8718cea863c410VgnVCM1000000b205a0aRCRD&vgnextchannel=00b3cf7588c97610VgnVCM2000001f4a900aRCRD.

Martuzzi, M., Mitis, F., Iavarone, I., Serinelli, M., 2006. Health impact of pm10 and ozone in 13 Italian cities. WHO Regional Off. Europe 133.

Marécal, V., Peuch, V.H., Andersson, C., Andersson, S., Arteta, J., Beekmann, M., Benedictow, A., Bergström, R., Bessagnet, B., Cansado, A., Chéroux, F., Colette, A., Coman, A., Curier, R.L., Denier van der Gon, H.a.C., Drouin, A., Elbern, H., Emili, E., Engelen, R.J., Eskes, H.J., Foret, G., Friese, E., Gauss, M., Giannaros, C., Guth, J., Joly, M., Jaumouillé, E., Josse, B., Kadygrov, N., Kaiser, J.W., Krajsek, K., 2015. A regional air quality forecasting system over Europe: the MACC-II daily ensemble production. Geosci. Model Dev. (GMD) 8, 2777–2813. https://doi.org/10.5194/gmd-8-2777-2015 (publisher: Copernicus GmbH).

de Medrano, R., Aznarte, J.L., 2020. On the Inclusion of Spatial Information for Spatio-Temporal Neural Networks. Neural Computing and Applications. In press.

Navares, R., Aznarte, J.L., 2020. Predicting air quality with deep learning LSTM: towards comprehensive models. Ecol. Inf. 55, 101019 https://doi.org/10.1016/j.ecoinf.2019.101019.

Nebenzal, A., Fishbain, B., 2018. Long-term forecasting of nitrogen dioxide ambient levels in metropolitan areas using the discrete-time Markov model. Environ. Model. Software 107, 175–185. https://doi.org/10.1016/j.envsoft.2018.06.001.

Nelsen, R.B., 1999. An introduction to copulas. In: Lecture Notes in Statistics. Springer-Verlag, New York. https://doi.org/10.1007/978-1-4757-3076-0. https://www.springer.com/gp/book/9781475730760.

Ouyang, W., Guo, B., Cai, G., Li, Q., Han, S., Liu, B., Liu, X., 2015. The washing effect of precipitation on particulate matter and the pollution dynamics of rainwater in downtown Beijing. Sci. Total Environ. 505, 306–314. https://doi.org/10.1016/j.scitotenv.2014.09.062.

Özkaynak, H., Glenn, B., Qualters, J.R., Strosnider, H., Mcgeehin, M.A., Zenick, H., 2009. Summary and findings of the epa and cdc symposium on air pollution exposure and health. J. Expo. Sci. Environ. Epidemiol. 19, 19–29.

Sellier, Y., Galineau, J., Hulin, A., Caini, F., Marquis, N., Navel, V., Bottagisi, S., Giorgis-Allemand, L., Jacquier, C., Slama, R., Lepeule, J., 2014. Health effects of ambient air pollution: do different methods for estimating exposure lead to different results? Environ. Int. 66, 165–173. https://doi.org/10.1016/j.envint.2014.02.001.

Sims, C.A., 1980. Macroeconomics and reality. Econometrica: J. Econometric Soc. 1–48.

Thatcher, M., Hurley, P., 2010. A customisable downscaling approach for local-scale meteorological and air pollution forecasting: performance evaluation for a year of urban meteorological forecasts. Environ. Model. Software 25, 82–92. https://doi.org/10.1016/j.envsoft.2009.07.014.

Vega García, M., Aznarte, J.L., 2020. Shapley additive explanations for NO2 forecasting. Ecol. Inf. 56, 101039 https://doi.org/10.1016/j.ecoinf.2019.101039.

Chapter 6

Side project I: A New Spatio-Temporal Neural Network Approach for Traffic Accident Forecasting

Type:	Published article
Journal:	Applied Artificial Intelligence
Authors:	Rodrigo de Medrano & José Luis Aznarte
Published:	August 2020
Impact factor:	1.580
5-Year Impact factor:	1.242
Quartile:	Q4 (Artificial Intelligence)
DOI:	10.1080/08839514.2021.1935588
Contribution:	Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing.

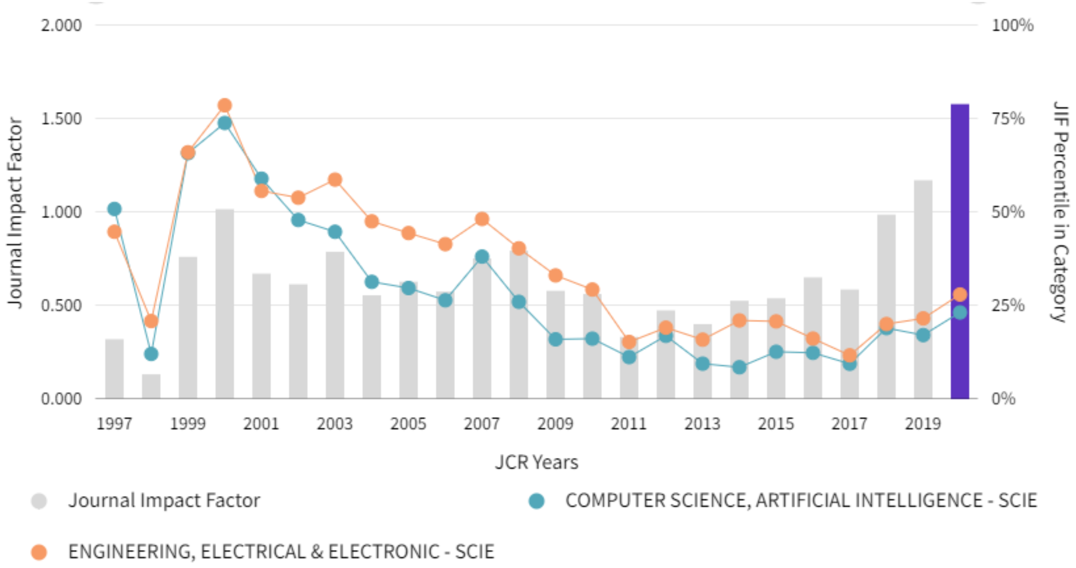


FIGURE 6.1: Impact factor: Applied Artificial Intelligence.

APPLIED ARTIFICIAL INTELLIGENCE
https://doi.org/10.1080/08839514.2021.1935588

Taylor & Francis
Taylor & Francis Group

Check for updates

A New Spatio-Temporal Neural Network Approach for Traffic Accident Forecasting

Rodrigo de Medrano and José L. Aznarte

Artificial Intelligence Department, Universidad Nacional de Educación a Distancia — UNED, Madrid, Spain

ABSTRACT

Traffic accidents forecasting represents a major priority for traffic governmental organisms around the world to ensure a decrease in life, property, and economic losses. The increasing amounts of traffic accident data have been used to train machine learning predictors, although this is a challenging task due to the relative rareness of accidents, interdependencies of traffic accidents both in time and space, and high dependency on human behavior. Recently, deep learning techniques have shown significant prediction improvements over traditional models, but some difficulties and open questions remain around their applicability, accuracy, and ability to provide practical information. This paper proposes a new spatio-temporal deep learning framework based on a latent model for simultaneously predicting the number of traffic accidents in each neighborhood in Madrid, Spain, over varying training and prediction time horizons.

ARTICLE HISTORY
Received 18 September 2020
Accepted 19 October 2020

Introduction

Nowadays, the urbanization trend around the globe has introduced new opportunities and issues in the cities. One of the most important aspects of modern society is related to the use of motorized vehicles as a method of transport. Although very efficient in several ways (Litman 2009), motor vehicles imply problems related to traffic and health care. For example, pollution and traffic accidents are some of the principal causes of death in cities all over the world (Kelly and Fussell 2015; WHO 2015).

This is the reason why the scientific interest for traffic accidents has increased in the past decades, and proposing solutions is a crucial issue for the sake of improving transportation and public safety. Being capable of understanding and reducing accidents has become an important commitment in many cities, as they not only cause significant life losses, but also property and economic ones (Peden et al. 2004).

In this work, an effort will be put to study the traffic accident phenomenon in the city of Madrid, Spain. This has been the subject of several lines of

CONTACT José L. Aznarte jlaznarte@dia.uned.es Artificial Intelligence Department, Universidad Nacional de Educación a Distancia — UNED, Madrid 28041, Spain

research in the past, although most previous studies on traffic accident prediction conducted by domain researchers simply applied classical prediction models on limited data without addressing many challenges properly, thus leading to unsatisfactory performances. For instance, the imbalanced severity classes, nonlinear relationship between dependent and independent variables, or spatial heterogeneity are usual problems to deal with in order to improve previous results in the field. Even with an accurate and complete statement of the problem, human and external factors (roads, vehicles, etc.) make this field highly challenging (Hoel et al. 2011; Vaa, Penttinen, and Spyropoulou 2007).

Although predicting the exact space-temporal position of accidents is out of the scope with actual techniques due to its complexity (Mannering and Bhat 2014; Zhang, Yau, and Chen 2013), much progress might be done by characterizing important parts of the problem. Trying to reduce the dimensionality of the space as much as possible, discovering relevant features or improving previous models are some examples of what can be done to provide insight in this particular problem.

In this context, this work presents the problem as a spatio-temporal series in which traffic intensity and meteorological variables play a central rol in predicting values for the traffic accidents series. For this purpose, we propose a new model, called XSTNN (from Exogenous Spatio-Temporal Neural Network) that consists of a deep learning approach for traffic accident regression based on spatio-temporal data. The model, which extends the Spatio-Temporal Neural Network (STNN) proposed by Delasalles et al. (2019) through the addition of external variables, is based on partitioning space into grid cells and taking advantage of the spatial relations existing in the series. A number of urban and environmental variables such as traffic intensity, rainfall, temperature, and wind are collected and map-matched with each grid cell. Given the number of accidents as well as the other urban and environmental features at each location, we learn a model to forecast the number of accidents that will occur in each grid cell in future timesteps.

By presenting the number of traffic accidents as a spatio-temporal series and learning how to model it, it is possible (for example) to increase emergency service's response time, focus the efforts to avoid potential dangers, create real-time safe routes recommendation systems, and, in short, reduce the losses that were discussed above. To the best of our knowledge, this is the first work that tackles the traffic accident forecast problem in the city of Madrid, although the proposed framework can be easily extend to any particular zone.

The rest of the paper is organized as follows: related work is discussed in Section 2, while Section 3 presents our datasets and the problem formulation. Section 4 introduces our deep learning model for traffic accident regression and Section 5 illustrates the evaluation of the proposed architecture as derived after appropriate experimentation. Finally, in Section 6 we point out future research directions and conclusions.

Related Work

Although very much studied, traffic accidents have been treated mostly in a "classical" context, by simply using statistical analysis in an attempt to understand better the phenomenon and the circumstances surrounding them. Examples that illustrate this situation can be found in Abdel-Aty and Radwan (2000), Lord (2006), and Roshandeh, Agbelie, and Lee (2016). There also are several works dealing with these methodologies and their typical issues (as for example Lord and Mannering 2010; Mannering and Bhat 2014). A long list of studies tackle the issue from the severity of the injuries perspective. Within this last group, de Oña, Mujalli, and Calvo (2011, 2013), Galatioto et al. (2018), Meysam et al. (2015), and Qiu et al. (2014) are some examples. Although instructive, most of these previous research fail to be able to apply all this knowledge to predict future events.

In a closer line to our work, during the last decade a considerably number of Artificial Intelligence-based approaches have appeared, taking advantage of the large datasets which are available nowadays. We can cite Chen (2017), Li et al. (2008), Lin, Wang, and Sadek (2015), and Zhang et al. (2018) as examples. As a first glance in the matter, these works provide new tools for solving the problem, but they lack relevant information in their analysis and mainly focus in showing the better performance of an specific model, without deepening in the behavior of their algorithms. However, there are some counterexamples like Tarek and Walid (1998), where a fair comparison is made between neural and fuzzy models in the field of traffic accident. In order to get more sophisticated and precise systems, last researches focus their efforts in new models as Variational Autoencoders, Deep Neural Networks, and video-based models for detecting and understanding better traffic accidents (Singh and Mohan 2019; Yu, Xu, and Gu 2019; Zheng et al. 2019).

Until now, the references presented here did not tackle the regression problem or were all lumped under the same hypothesis: ignoring the importance of the spatial dimension in the traffic accidents forecasting. However, a number of studies have pointed out how relevant this variable is in order to get appropiate results (Xu and Huang 2015; Rhee et al. 2016). Since then, more and more researches focus their efforts in the spatio-temporal (and not just temporal) prediction problem. We can cite Ren et al. (2017), Yang, Wang, and Yu (2018), and Yuan, Zhou, and Yang (2018) as some of the most relevant works, some of them being classified under the label of Deep Learning. Specifically, some of these last references point at exogenous variables as helpful in the forecast process.

While traffic accident research from an Artificial Intelligence perspective *per se* is still a young field, its importance makes them be a central variable of a vast number of Intelligent Transport Systems studies. For example, several of them in which routes recommendation systems and vehicle routing problems are stated, identify traffic accidents as potential variables that might have direct impact in the system. Is the case of Eshtehadi, Demir, and Huang (2020), the routing problem is tackled by an adapted adaptive large neighborhood search algorithm. Similarly, Du et al. (2019) focus their efforts in solving a similar problem but trying to minimize the total transportation risk, time and cost by using an improved biogeography-based optimization algorithm. As the authors point out, when transporting hazardous materials traffic accidents might suppose a great risk. In Salman and Alaswad (2018), a model based on Markov chain for traffic optimization to decrease congestion is presented. In those cases, although unpredicted traffic pattern changes are contemplated, a robust traffic accident forecasting system could be beneficial. In the same way, Sumit and Akhter (2019) show that traffic accidents might be a main actor for road weight calculation. By using a c-means clustering and deep-neurofuzzy model, they use (among other variables) real-time accident data for detecting traffic congestion, monitoring traffic status, and deciding optimum route. Another example can be found in Nasri, Bekta¸s, and Laporte (2018), where it is showed another field that could benefit from traffic accident research: autonomous vehicles and their anticipation to potentially dangerous situations. Cunneen et al. (2019) show how Artificial Intelligence could also contribute in the ethical aspects of this matter. In this same context, Liu et al. (2016) propose V2I communications between vehicles in order to improve traffic condition. Particularly, an accident prediction system could refine future vehicle's decisions.

Problem Formulation and Data

Problem Formulation

Given a spatial grid S, where each grid is represented as s_i, and a timestep t_j, we aim to learn a model to predict the number of accidents in each grid s_i during each time slot t_j. This mean that a spatio-temporal sample writes as $x(s_i; t_j) : j = 1, \dots, T; i = 1, \dots, S$.

Although spatial zones might be defined arbitrarily, it is expected that using intrinsic spatial information could be helpful. More precisely, we propose that each grid s_i represents a neighborhood of Madrid as it is expected that each neighborhood presents different peculiarities that might be related to traffic accidents. Moreover, we use an hour as the length of our timestep t_j. Without loss of generality, other values could be chosen for s_i and t_j. We work with data from year 2018 for both the training and validation sets. Only in-city accidents are treated, as road accidents present different peculiarities.

For the rest of the section, all the data cleaning and manipulation will take into consideration this proposed framework.

Data Sources

- **Traffic accident data**: Provided by *Portal de datos abiertos del Ayuntamiento de Madrid*,[1] it summarizes all the information related to car crashes in the city of Madrid. Specifically, for every accident it shows physical location (although not geographycal), date (year, month, and day), time (hour), sex and severity for each person involved and several meteorological conditions. The last two variables of this dataset were not taken in consideration, as they were not relevant or there were better sources for them (concretley weather data later in this same section). For example, sex can be relevant when making statistics of the phenomena, but irrelevant when trying to predict new accidents.

Spatial information is presented as city addresses (street and number or intersection), while temporal information is limited to the hour in which the accident was reported.

- **Traffic data**: As before, provided by *Portal de datos abiertos del Ayuntamiento de Madrid*. This dataset contains historical data of traffic measurement points in the city of Madrid. The measurements are taken every hour at each point, including traffic intensity in number of cars per hour and average speed in m/s. Some other traffic parameters, although unused in this project, are present in this set too.

Spatial information is given with the coordinates (longitude and latitude) of measurement points, while temporal information is taken every 15 minutes.

- **Weather data**: Weather data were provided by the *Red Meteorológica Municipal*.[2] Weather observations consist of hourly temperature in Celsius degrees, solar radiation in W/m2, wind speed measured in ms-1, wind direction in degrees, daily rainfall in mmh-1, pressure in mbar, degree of humidity in percentage, and ultraviolet radiation in mWm-2 records.

Weather information is taken along six different stations. It is reported hourly.

Some cleaning work was necessary to work through the data. It is worth noting that these decisions are fundamental as error might be introduced in the system during this cleaning process.

Firstly, Google Maps Api[3] was used for geocoding the adresses provided in the dataset.

With respect to traffic intensity, it is worth pointing out that is the only set that does not present its information hourly, but every 15 minutes. In order to have a final homogeneus dataset, average over every entire hour is calculated. Note that typical deviation of traffic intensity over and hour represents less than 10% of the real values on average. In addition, the average of the traffic intensity is taken for each neighborhood as if every measurement point was a different sample from the same phenomenon for every zone. Once more, the standard deviation that results from this decision is less than 5% respect to the mean, showing that there is a predisposition to have similar traffic conditions for each neighborhood.

Finally, while the actual meteorological data were taken in six substations in the city of Madrid, our own data consist of average hourly variables from those six substations. Although this decision could be seen as a loss of information, this approximation is enough for a first insight. Also, assigning different meteorological variables for each accident depending on its location supposes an extra difficulty when using a spatial mesh (the six substations) different from the one used in this work (neighborhoods of Madrid).

Data Analysis

Through this section, we will explore if our data can be modeled as a spatio-temporal series. This will be done by an exploratory analysis for both dimensions.

To explore its time dependency, it is possible to use a boxplot of different time windows of traffic accident count for different time periods of Madrid as in Figure 1. Clearly, the traffic accident patterns change drastically for different time periods. Specifically, traffic accidents are more frequent at traffic rush hours than that at off-peaks, on weekdays than on weekends, and they reflect a decrease in summer holiday days. This figure reveals some characteristical periodicities that expose a hidden time dependence in traffic accidents, letting us model the series as a temporal one.

To determine whether the number of traffic accidents is associated with the spatial location, the heatmap of number of traffic accidents is plotted for Madrid in 2018 (Figure 2). As it can be seen, the number of traffic accidents is not uniformly distributed, and it is highly related with the geographical position of a neighborhood. Usually, the neighborhoods with highest traffic accident concentrations lie in the major commercial and business areas.

From this two last figures, we can point out one of the special difficulties of the traffic accidents series: how infrequent accidents are. In this context, and from the frequentist probability point of view, the odds of an accident taking place anytime in an hour and at any neighborhood is about 0.8%.

Deep Model for Traffic Accident Forecasting

This paper presents a new deep learning neural model which is based on the work from Ziat *et al.* (Delasalles et al. 2019). Specifically, they introduced a method for spatio-temporal series forecasting problems, such as meteorology, oceanography, or traffic, formalized as a recurrent neural network for modeling time-series of spatial processes. Our model preserves this nature but it is an improvement from the point of view of its usability, allowing us to make use of external (or exogenous) variables. Concretely, the model learns these spatio-

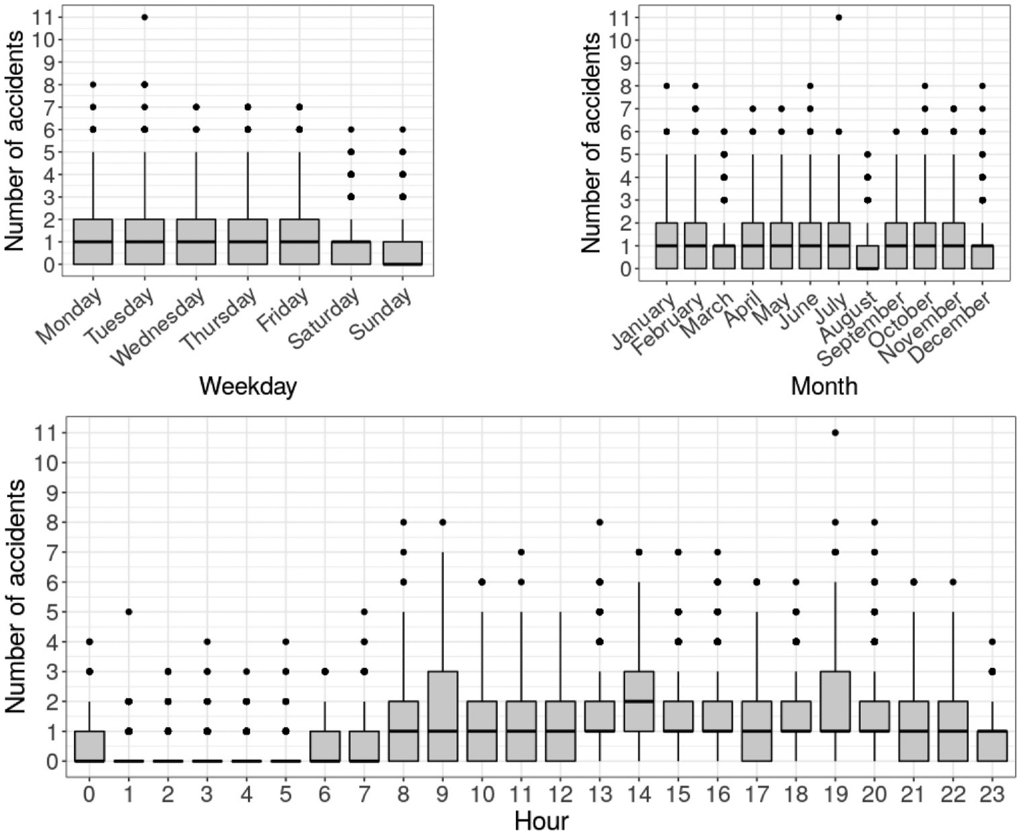


Figure 1. Periodicities of the traffic accidents series. (a) Number of accidents depending on day of the week. Weekends present less number of accidents. (b) Number of accidents for each month. August seems to be safer. (c) Number of accidents depending on hour of the day. In this case we have the most clear difference.

temporal dependencies through a structured latent dynamical representation, while a decoder predicts the observations from the latent space.

Notation

Let us first introduce the notation that will be used througout this chapter. Denoting n as the number of series, T their length and m the dimensionality of them. In our specific domain, there will be as many series (n) as spatial zones. Moreover, $m = 1$ as every series will be composed of only one dimension: traffic accidents.

If we call X as the values of all the series between instants 1 and T, then X is a tensor in $\mathbb{R}^{T \times n \times m}$. At last, $X_t \in \mathbb{R}^{n \times m}$ is a tensor that denotes the values of all the series at time t.

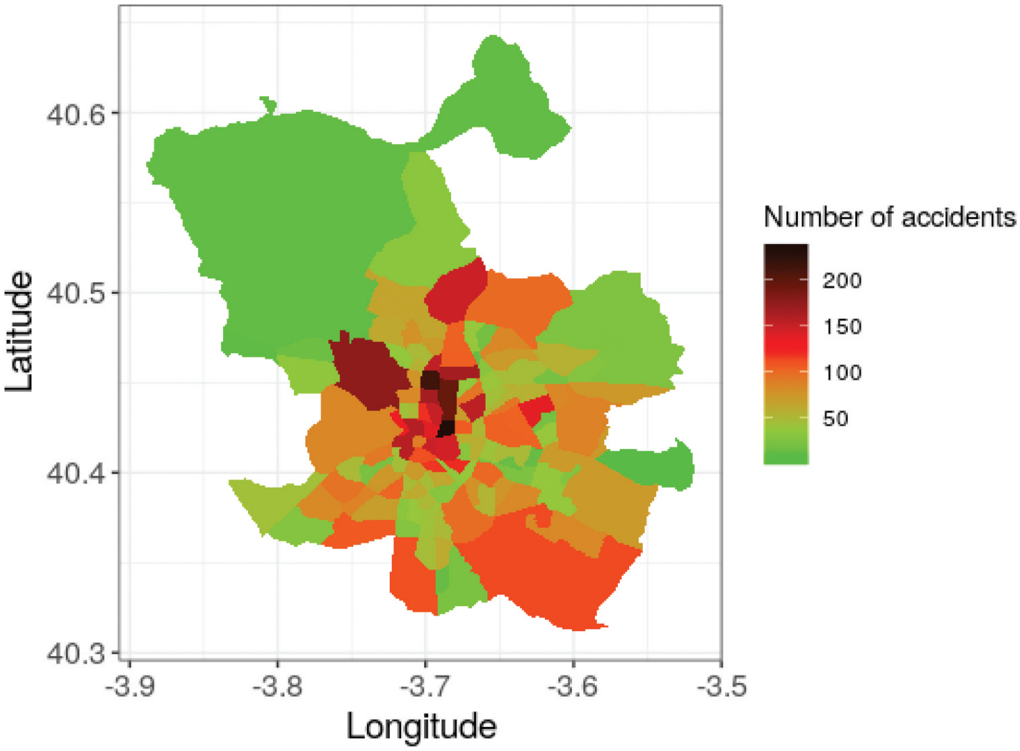


Figure 2. Total number of accidents by neighborhoods of Madrid during 2018.

The STNN Model

Let Z_t be the latent representation, or latent factors, of the series at time t. The model has two principal components: the dynamic function (denoted as g), and the decoder function (called d). The first one is in charge of controlling the dynamics of the system, calculating the next latent state based on the previous one: $Z_{t+1} = g(Z_t)$. The second one is a decoder which maps latent factors Z_t onto a prediction of the actual series values at time t: $\tilde{X}_t = d(Z_t)$, $\tilde{X}_t$ being the prediction computed at time t.

As it should be clear, the parameters of both functions (g and d) are learned so that the essence of the series is captured. Unlike usual neural networks, the latent representation Z_t is treated as a parameter too, distinguishing this model and making it more flexible than usual recurrent neural networks.

The idea behind the spatial component is to consider each zone as a different series with its own latent representation at each time step. For a latent space dimension of N, Z_t is a $n \times N$ tensor such that $Z_{t,i} \in \mathbb{R}^N$ is the latent factor of series i at time t. Thus, we have the following relations:

$$d : \mathbb{R}^{n \times N} \longrightarrow \mathbb{R}^{n \times m} \tag{1a}$$

$$g : \mathbb{R}^{n \times N} \longrightarrow \mathbb{R}^{n \times N} \tag{1b}$$

Not only each spatial zone has a series, spatial information is integrated in the dynamic component of the model through a matrix $W \in \mathbb{R}_{+}^{n \times n}$ that shares information between all the zones. Although this matrix will be provided, the actual model is also capable of learning it.

The latent representation of each series at time $t + 1$ depends on the previous state of all the series (included itself). Hence, we can separate the calculation of a new state by two different sources: intra-dependency in the first term of the right-hand side of (3) and inter-dependency in the second term. The first one aims to get the dynamic of each series as an individual entity, whereas the second one is devised to exploit spatial relations between all series. This way, the model considers a different temporal series in each spatial zone while keeping information about the spatial relation between all of them. Formally, the dynamic model $g(Z_t)$ is designed as follows:

$$Z_{t+1} = h(Z_t \Theta^{(0)} + W Z_t \Theta^{(1)}) \tag{2}$$

In this last equation, h is a nonlinear function ($h = tanh$ in this project) and Θ denotes a parametrized function $\Theta \in \mathbb{R}^{N \times N}$. In this case, Θ will be a linear function or a multilayer perceptron (MLPs), although could be any parametrized function.

Including Exogenous Variables: The XSTNN Model

The main limitation of the STNN is that it is not able to take into account the exogenous variables which might be related to the process being modeled and which could enrich the internal representation and, thus, improve the predictions. The XSTNN aims to resolve this.

Let us denote the exogenous variables Λ. The main idea will be to change equation 2 so that the latent space is modified directly by Λ. These variables are temporal series, so they can be treated on the same way we did previously, meaning that Λ_t denotes the slice of Λ at time t. Due to the possibility of using several exogenous variables, Λ_t is a $n \times m$ tensor.

By introducing Λ in the estimation of Z_t, the model learns the dynamics taking into account external information too. As the premise of this work is to assume that exogenous variables might change the dynamic of the series, learning to mold the system in function of both meets our requirements the best.

Once the main idea has been explained, it is necessary to answer some other questions. Concretely, there are a few alternatives for reconstructing (2) in the way it was intended. Moreover, a discussion about what time step to use with Λ is desirable: when computing Z_t, both Λ_t and Λ_{t+1} might be beneficial. The first one represents the idea of a previous state having an effect on the next one, whereas the second option symbolizes the conception of an actual state modifying the series.

Let us now introduce some possibilities. First, if exogenous data does not present spatial dependency, it can be more efficient to avoid the use of spatial relations for Λ. This version writes:

$$Z_{t+1} = h(Z_t\Theta^{(0)} + WZ_t\Theta^{(1)} + \Lambda_t\Theta^{(2)}) \tag{3}$$

On the contrary, when exogenous variables may exhibit spatial dependency, the same treatment that Z has will be provided to Λ. This notion is captured as follows:

$$Z_{t+1} = h(Z_t\Theta^{(0)} + WZ_t\Theta^{(1)} + \Lambda_t\Theta^{(2)} + W\Lambda_t\Theta^{(3)}) \tag{4}$$

A diagram that represents this last option is presented in Figure 3.

Overall, the model is similar to the STNN. Both the optimization problem and the training (loss function, learning algorithm, inference, etc.) are applicable to the XSTNN model.

However, we would like to point out the two principal limitations of our proposal:

- Using an specific matrix W for a concrete problem means that, for different circumstances (for example, a different spatial grid), a retraining is needed.
- Both the dynamic and the decoder functions are stationary, meaning that it do not change over time. In Delasalles et al. (2019) a method to tackle this problem is proposed.

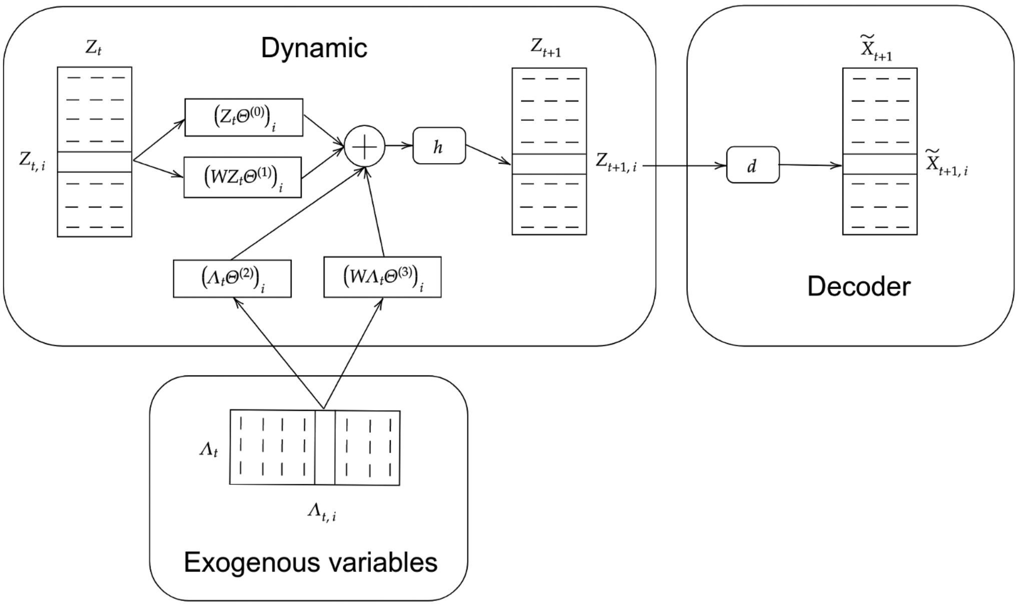


Figure 3. Architecture of the XSTNN model as described in Sect. 4.3.

Experimental Results

Before explaining the experiments, we will establish what questions we wish to answer. They are stated as follow: (1) Are the results of the proposed model better when compared with benchmark methods, including classical predictive models, tree-based models and STNN? (2) Is our proposed model capable of managing different spatial regions or timesteps? (3) Do the forecasting results make sense? Does our model provide more insights on the problem? (4) Are the predicted accident locations correlated with the ground truth spatially?

Through these questions, we expect to evaluate if the XSTNN model supposes a step forward in the prediction of traffic accidents.

Baselines Models and Evaluation Metrics

Several methods have been chosen to be compared with the XSTNN. Concretely, the STNN itself, a XGBoost tree-based algorithm (Chen and Guestrin, 2016), linear regression, and a naive mean and persistence models. The mean model forecasts new values of the series using the mean of past values from the same series, while persistence model uses the last value for each series for making the prediction.

To evaluate the accuracy and precision of the prediction, we selected Mean Absolute Error (MAE) and Bias as our metrics. In a spatio-temporal context (Wikle, Zammit-Mangion, and Cressie 2019), they are defined as:

$$\mathrm{MAE} = \frac{1}{TS}\sum_{j=1}^{T}\sum_{i=1}^{S}\left|x_{s_i;t_j} - \tilde{x}_{s_i;t_j}\right| \tag{6}$$

$$\mathrm{Bias} = \frac{1}{TS}\sum_{j=1}^{T}\sum_{i=1}^{S}\left(x_{s_i;t_j} - \tilde{x}_{s_i;t_j}\right) \tag{7}$$

where, as it was defined in Section 2, $x(s_i;t_j) : j = 1,\ldots,T; i = 1,\ldots,S$ is a spatio-temporal sample from the real series, $\tilde{x}(s_i;t_j)$ makes reference to the predicted series, S is the total number of spatial grids and T the total number of timesteps.

Performance Evaluation

To validate the different proposed methodologies, a time series cross-validation scheme called rolling origin is used (Tashman 2000). Rolling origin is an evaluation technique according to which the forecasting origin is updated successively and the forecasts are produced from each origin. This technique allows obtaining several forecast errors for time series, which gives a better understanding of how the models perform.

Let us now describe how the previous procedure is applied in our own experiments. Consider the following steps:

(1) The traffic accidents dataset is splitted in ten succesive sets, that is to say, starting all sets from January, 1st of 2018 at 00:00, each of those ten sets end at a different date between February, 14th at 23:00 and December, 31st at 18:00. To consider, all datasets are equally spaced and a minimun of 45 days have been set for training.
(2) As a test set, we consider predictions within a 5 hour horizon. For a train set of T timesteps, this means that the evaluation of the quality of the model will be made over $T+1$ to $T+5$ timesteps.
(3) Finally, the ten splitted sets are trained and validated over $T+1$ to $T+5$ timesteps. The final error is the average of all validations. The datasets have been chosen with the purpose that different hours and week days are tested for a more complete and extensible validation.

This procedure is equivalent for all models. It was applied to both the parameter tuning and the final training process.

Experimental Setup and Parameter Tuning

We set up the neural networks experiments and the other two models on a external machine proportionated by *Departamento de Inteligencia Artificial, UNED* .[4] The STNN and the XSTNN[5] were built upon PyTorch. Concretely, an early-stopping approach using Adam optimizer with the settings: $\beta_1 = 0.0$, $\beta_2 = 0.999$, $\in = 10^{-9}$ and $w_d = 10^{-6}$ was used for both methodologies. The mean, persistence, linear regression, and XGboost models are built on R, the last one made use of the package *xgboost* (Chen and Guestrin, 2016).

With respect to parameter and hyper-parameter tuning, we grid-searched hyper-parameters on each models for the sake of achieving the best possible results. The final hyper-parameters used for this work are gathered in Table 1.

Any other hyper-parameter not taken into account in this tuning process, are used with their default values. We decided to set matrix W (spatial relations, introduced in Section 3), as the inverse of spatial distance. Thus, all zones are in some way related but in a bigger degree the closer they are. Lastly, each series was rescaled between 0 and 1.

Results and Discussion

In order to identify quantitatively the performance of the different models and baselines, Table 2 provides the average prediction error for $T+1$ to $T+5$. From this first insight it should be clear that both XSTNN and STNN

Table 1. Values used for each hyper-parameter. n_z is the dimension of the latent space. The remaining variables were presented in Section 3 or are commonly used parameters.

STNN	Learning rate	0.01
λ	0.01	
n_z	2	
$g(Z)$	Linear	
Minibatch size	512	
Dropout	0.25	
XSTNN	Learning rate	0.01
λ	0.1	
n_z	2	
$g(Z)$	Linear	
Minibatch size	512	
Dropout	0.35	
XGBoost	Number of rounds	80
Max. depth	15	
η	0.1	
γ	1	
Min. child weight	1	
Subsample	0.7	

Table 2. Performance for $T+1$ to $T+5$ traffic accident regression.

Model	MAE	Bias
XSTNN	**0.0041 ± 0.0006**	− 0.0006 ± 0.0004
STNN	0.0045 ± 0.0006	− 0.0004 ± 0.0006
XGBoost	0.0052 ± 0.0006	0.0004 ± 0.0006
Linear regression	0.0050 ± 0.0006	0.0002 ± 0.0007
Mean	0.0052 ± 0.0007	0.0003 ± 0.0007
Persistence	0.0055 ± 0.0008	0.0006 ± 0.0007

outperform the other models. As Mean model and XGboost were trained taking into account the existence of a spatial grid but without establishing relations between them, these results confirm that making use of prior spatial information is beneficial for the regression problem. Beyond that, the XSTNN presents a better performance than the STNN.

For a more detailed vision, Figure 4 shows the distribution of the metrics and the average error by timestep. From this figure, same conclusions can be extracted as before: the XSTNN model presents a better general behavior compared to the rest of the models. Again, the fact of introducing spatial knowledge to the problem stands as an appropriated approach for this particular series, and our results reinforce the idea that introducing exogenous variables is favorable for the regression problem. However, it is worth noting that there is not a clear relation between errors and timestep. Although an increment on the error by timestep in the prediction is usually expected (cumulative error), the randomness of traffic accidents does not let us extract clear conclusions from this aspect.

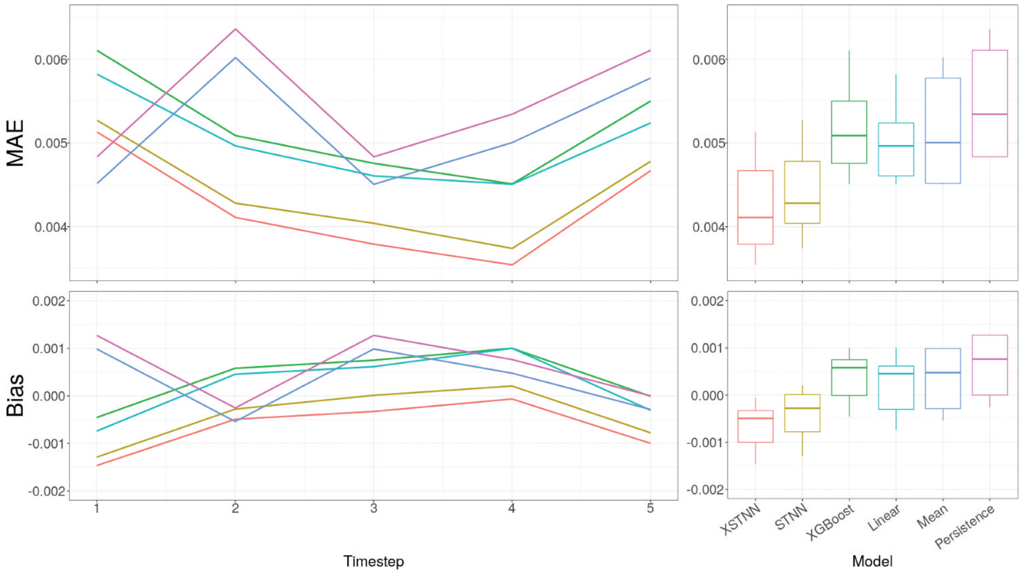


Figure 4. Forecasting performance (MAE and bias) of the different models by timestep together with the calculated distributions.

Beyond the quantitative analysis, now we show some accomplishments from our proposed model respect to the STNN. For that purpose, we will take a deeper look into a concrete example, without loss of generality.

Let us introduce the following situation: we forecast the accident regression series from 17 p.m. to 21 p.m. on a Wednesday. From Figure 1 we know this situation corresponds to a high-risk circumstance for traffic accidents to happen. In this context, Figure 5 illustrates a comparison of our two principal

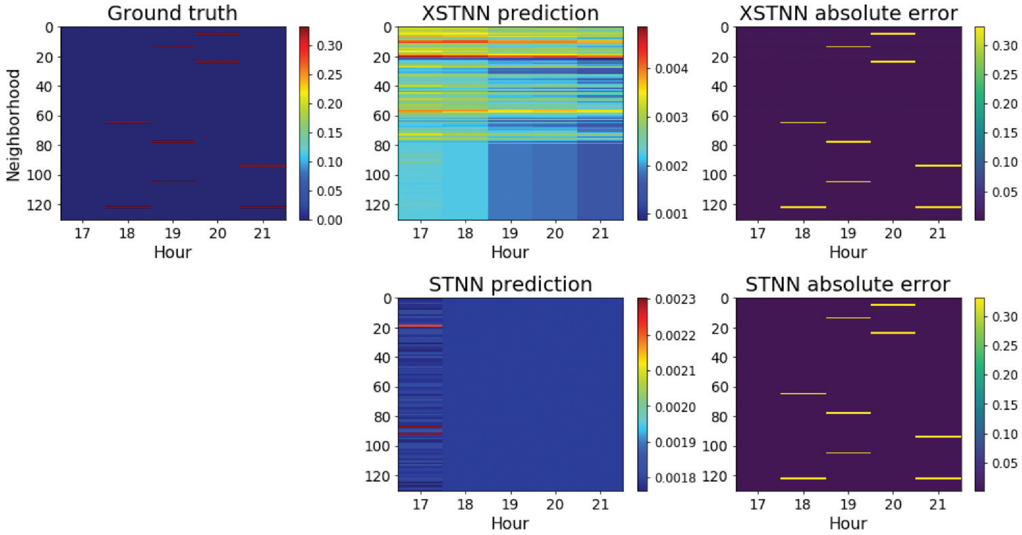


Figure 5. A practical example of the operation of both networks, XSTNN and STNN, for the same situation. From 17 p.m. to 21 p.m. on a Wednesday.

models with a levelplot (time in x axis, neighborhoods in y axis and colored by traffic accidents). Let us expose several ideas.

First of all, and unfortunately, the regression problem is far from being solved. A comparison of colorbars from both, STNN and XSTNN predictions, with the ground truth corroborates this statement. As Chen et. al. have documented, after some analysis of traffic accident data, it is difficult to predict whether traffic accidents will happen or not directly, because complex factors can affect traffic accidents, and some factors, such as the distraction of drivers, cannot be observed and collected in advance (Chen et al. 2016). Nevertheless, our XSTNN model has proved to be a new step in the right direction, outperforming the rest of baselines models (Table 2).

Secondly, the next natural question that rises is about the reason of this improvement. Again, Figure 5 sheds light on this matter. Whereas the STNN quiclky truncates its values close to 0 for every neighborhood and timestep, the XSTNN takes some risks and it is able to differentiate between time intervals and spatial zones. As the most likely situation is having no accidents for each hour and neighborhood, both networks have values approaching to 0 as outputs.

Certainly, taking more risks does not ensure a better performance in the regression problem. It is necessary that the model manages to elucidate which time intervals and neighborhoods are more important for the problem that we have in hand as a function of past events. In this concrete case, the model has learned to prioritize neighborhoods from 1 to 80, as they report a vast majority of the total number of traffic accidents in the city of Madrid. Besides, the XSTNN reveals a negative trend over the hours as we would expect.

As XSTNN learns better to distinguish between time ranges and spatial zones, it is possible to find other situations in which, again, this model offers more information and assimilates the system's dynamics in a better way. For example, and to corroborate that the XSTNN behaves better in a variety of situations, Figure 6 gives evidence of a totally different state on a Sunday from 6 a.m. to 10 a.m. In this context, we will expect a higher risk at last late hours and at past 9 a.m., the XSTNN correspondingly adapting its output to this situation. On the contrary, the STNN is not capable of learning the corresponding dynamic. Unlike previously (Figure 5), this time the XSTNN takes less risks and its output is closer to 0 as we would expect less accidents on a Sunday morning that a Wednesday on the evening as before.

Through the previous discussion we have pointed out how the XSTNN infers properties based on the time condition and the concrete spacial zone. For this last case, Figure 7 offers an analysis of spatial risk for each neighborhood. Both series, the real and the predicted ones, were rescaled for a direct comparison between them. This way, it is clear that the XSTNN is capable of reasoning in both dimensions, temporal and spatial.

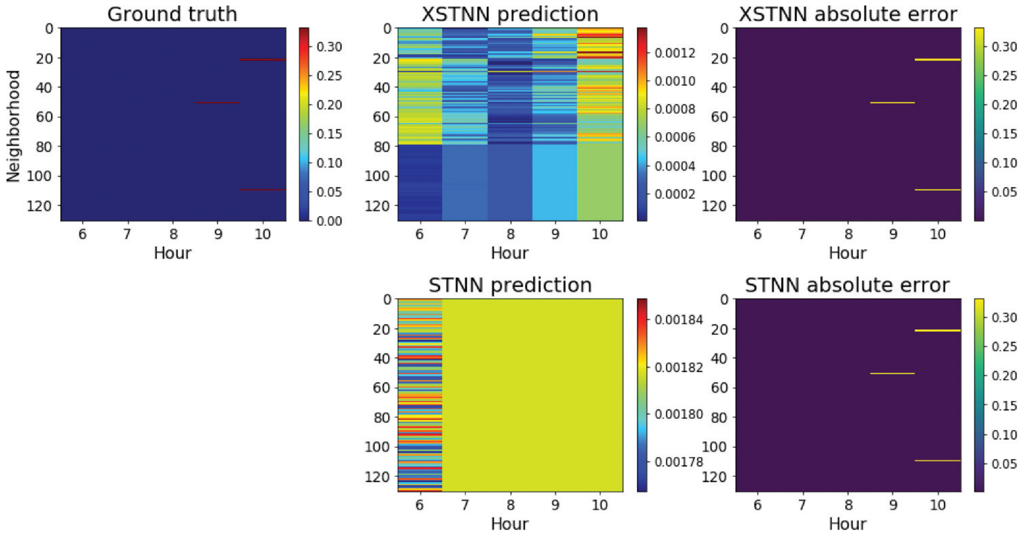


Figure 6. A practical example of the operation of both networks, XSTNN and STNN, for a same situation. From 6 a.m. to 10 a.m. on a Sunday.

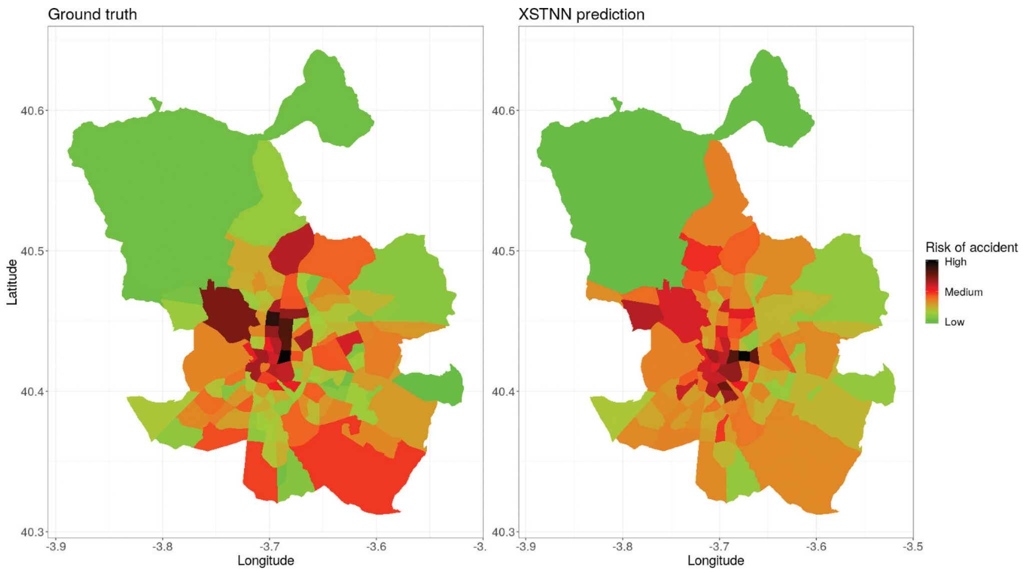


Figure 7. Spatial risk in the same scale for the ground truth (left) and the XSTNN (right).

In summary, the XSTNN reports a better understanding and learning of the dynamic of the system, being more flexible and creative in its prediction. These features translate into a better performance than their direct rivals.

Conclusions

Through this work, a new approach for spatio-temporal series forecasting called XSTNN has been proposed. The problem of traffic accidents prediction was tackled by this new neural network model, showing a better performance than the rest of baselines model. Also, the exposed model is easily extendable to any temporal or spatial configuration. Although traffic accidents regression is challenging due to several difficulties, the XSTNN has proved to stand out for its capability to provide a deeper insight in the problem series and to adapt its reasoning to a larger number of different situations. Thus, this paper demonstrates that spatio-temporal neural networks are a promising field for traffic accident prediction in the future.

Future work in this field can be extended to incorporate other features that are not necessarily series, like economics or demographics. Also, the XSTNN model might be extended by introducing more temporal terms from exogenous series for updating the latent space.

Notes

1. https://datos.madrid.es/portal/site/egob/
2. http://www.mambiente.madrid.es
3. https://cloud.google.com/maps-platform/
4. http://www.ia.uned.es/
5. Code available at https://github.com/rdemedrano/xstnn

Funding

This research was partially funded by the *Empresa Municipal de Transportes* (EMT) of Madrid under the program "*Aula Universitaria EMT/UNED de Calidad del Aire y Movilidad Sostenible*".

ORCID

Rodrigo de Medrano http://orcid.org/0000-0002-4428-7053

References

Abdel-Aty, M. A., and A. E. Radwan. 2000. Modeling traffic accident occurrence and involvement. *Accident Analysis & Prevention* 32 (5):633–42. doi:10.1016/S0001-4575(99)00094-9.

Chen, C. 2017. Analysis and forecast of traffic accident big data. In *ITM Web of Conferences 12*, 04029, Guangzhou (China).

Chen, Q., X. Song, H. Yamada, and R. Shibasaki. 2016. Learning deep representation from big and heterogeneous data for traffic accident inference. In AAAI: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. Phoenix, USA.

Chen, T., and C. Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, 785–94, San Francisco, USA.

Cunneen, M., M. Mullins, F. Murphy, and S. Gaines. 2019. Artificial driving intelligence and moral agency: Examining the decision ontology of unavoidable road traffic accidents through the prism of the trolley dilemma. *Applied Artificial Intelligence* 33(3):267–93. Publisher: Taylor & Francis eprint. doi:10.1080/08839514.2018.1560124.

de Oña, J., G. López, and J. Abellán. 2013. Extracting decision rules from police accident reports through decision trees. *Accident Analysis & Prevention* 50:1151–60. doi:10.1016/j.aap.2012.09.006.

de Oña, J., R. O. Mujalli, and F. J. Calvo. 2011. Analysis of traffic accident injury severity on spanish rural highways using bayesian networks. *Accident Analysis & Prevention* 43 (1):402–11. doi:10.1016/j.aap.2010.09.010.

Delasalles, E., A. Ziat, L. Denoyer, and P. Gallinari. 2019 December. Spatio-temporal neural networks for space-time data modeling and relation discovery. *Knowledge and Information Systems* 61(3):1241–67. doi:10.1007/s10115-018-1291-x.

Du, J., X. Li, L. Li, and C. Shang. 2019. Urban hazmat transportation with multifactor. *Soft Computing 24*, 6307–6328.

Eshtehadi, R., E. Demir, and Y. Huang. 2020 March. Solving the vehicle routing problem with multi-compartment vehicles for city logistics. *Computers & Operations Research* 115:104859. doi:10.1016/j.cor.2019.104859.

Galatioto, F., M. Catalano, N. Shaikh, E. McCormick, and R. Johnston. 2018 November. Advanced accident prediction models and impacts assessment. *IET Intelligent Transport Systems* 12(9):1131–41. doi:10.1049/iet-its.2018.5218.

Hoel, J., M. Jaffard, C. Boujon, and P. Van Elslande. 2011. Different forms of attentional disturbances involved in driving accidents. *IET Intelligent Transport Systems* 5 (2):120. doi:10.1049/iet-its.2010.0109.

Kelly, F. J., and J. C. Fussell. 2015. Air pollution and public health: Emerging hazards and improved understanding of risk. *Environmental Geochemistry and Health* 37 (4):631–49. doi:10.1007/s10653-015-9720-1.

Li, X., D. Lord, Y. Zhang, and Y. Xie. 2008. Predicting motor vehicle crashes using support vector machine models. *Accident Analysis & Prevention* 40 (4):1611–18. doi:10.1016/j.aap.2008.04.010.

Lin, L., Q. Wang, and A. W. Sadek. 2015. A novel variable selection method based on frequent pattern tree for real-time traffic accident risk prediction. *Transportation Research Part C: Emerging Technologies* 55:444–59. doi:10.1016/j.trc.2015.03.015.

Litman, T. A. 2009. Transportation cost and benefit analysis: Techniques, estimates and implications, Victoria Transport Policy Institute, 2nd ed. 1-19.

Liu, Y., J. Ling, Q. Wu, and B. Qin. 2016. Scalable privacy-enhanced traffic monitoring in vehicular ad hoc networks. *Soft Computing* 20 (8):3335–46. doi:10.1007/s00500-015-1737-y.

Lord, D. 2006. Modeling motor vehicle crashes using poisson-gamma models: Examining the effects of low sample mean values and small sample size on the estimation of the fixed dispersion parameter. *Accident Analysis & Prevention* 38 (4):751–66. doi:10.1016/j.aap.2006.02.001.

Lord, D., and F. Mannering. 2010. The statistical analysis of crash-frequency data: A review and assessment of methodological alternatives. *Transportation Research Part A: Policy and Practice* 44 (5):291–305.

Mannering, F. L., and C. R. Bhat. 2014. Analytic methods in accident research: Methodological frontier and future directions. *Analytic Methods in Accident Research* 1:1–22.

Meysam, E., R. M. Ali, H. Farshad, and S. Shahin. 2015 November. Prediction of crash severity on two-lane, two-way roads based on fuzzy classification and regression tree using geospatial analysis. *Journal of Computing in Civil Engineering* 29(6):04014099. doi:10.1061/(ASCE)CP.1943-5487.0000432.

Nasri, M. I., T. Bekta¸s, and G. Laporte. 2018. Route and speed optimization for autonomous trucks. *Computers & Operations Research* 100:89–101. doi:10.1016/j.cor.2018.07.015.

Peden, M., R. Scurfi, D. Sleet, D. Mohan, A. A. Hyden, and E. Jarawan. 2004. World report on road traffic injury prevention.

Qiu, C., C. Wang, B. Fang, and X. Zuo. 2014. A multiobjective particle swarm optimization-based partial classification for accident severity analysis. *Applied Artificial Intelligence* 28(6):555–76. Publisher: Taylor & Francis eprint. https://www.tandfonline.com/doi/pdf/10.1080/08839514.2014.923166

Ren, H., Y. Song, J. Wang, Y. Hu, and J. Lei. 2017. A deep learning approach to the citywide traffic accident risk prediction. In *2018 IEEE International Conference on Intelligent Transportation Systems (ITSC), Maui, Hawaii, USA*.

Rhee, K.-A., J.-K. Kim, Y.-I. Lee, and G. F. Ulfarsson. 2016. Spatial regression analysis of traffic crashes in seoul. *Accident Analysis & Prevention* 91:190–99. doi:10.1016/j.aap.2016.02.023.

Roshandeh, A. M., B. R. D. K. Agbelie, and Y. Lee. 2016. Statistical modeling of total crash frequency at highway intersections. *Journal of Traffic and Transportation Engineering (English Edition)* 3 (2):166–71. doi:10.1016/j.jtte.2016.03.003.

Salman, S., and S. Alaswad. 2018. Alleviating road network congestion: Traffic pattern optimization using Markov chain traffic assignment. *Computers & Operations Research* 99:191–205. doi:10.1016/j.cor.2018.06.015.

Singh, D., and C. K. Mohan. 2019. Deep spatio-temporal representation for detection of road accidents using stacked autoencoder. *IEEE Transactions on Intelligent Transportation Systems* 20 (3):879–87. doi:10.1109/TITS.2018.2835308.

Sumit, S. H., and S. Akhter. 2019. C-means clustering and deep-neuro-fuzzy classification for road weight measurement in traffic management system. *Soft Computing* 23 (12):4329–40. doi:10.1007/s00500-018-3086-0.

Tarek, S., and A. Walid. 1998 January. Comparison of fuzzy and neural classifiers for road accidents analysis. *Journal of Computing in Civil Engineering* 12(1):42–47. doi:10.1061/(ASCE)0887-3801(1998)12:1(42).

Tashman, L. J. 2000. Out-of-sample tests of forecasting accuracy: An analysis and review. *International Journal of Forecasting* 16 (4):437–50. doi:10.1016/S0169-2070(00)00065-0.

Vaa, T., M. Penttinen, and I. Spyropoulou. 2007 June. Intelligent transport systems and effects on road traffic accidents: State of the art. *IET Intelligent Transport Systems* 1(2):81–88. doi:10.1049/iet-its:20060081.

WHO. 2015. *WHO | data.*

Wikle, C. K., A. Zammit-Mangion, and N. Cressie. 2019. *Spatio-temporal statistics with R.* 1st ed. Chapman and Hall/CRC, London, United Kingdom.

Xu, P., and H. Huang. 2015. Modeling crash spatial heterogeneity: Random parameter versus geographically weighting. *Accident Analysis & Prevention* 75:16–25. doi:10.1016/j.aap.2014.10.020.

Yang, K., X. Wang, and R. Yu. 2018. A bayesian dynamic updating approach for urban expressway real-time crash risk evaluation. *Transportation Research Part C: Emerging Technologies* 96:192–207. doi:10.1016/j.trc.2018.09.020.

Yu, Y., M. Xu, and J. Gu. 2019. Vision-based traffic accident detection using sparse spatio-temporal features and weighted extreme learning machine. *IET Intelligent Transport Systems* 13 (9):1417–28. doi:10.1049/iet-its.2018.5409.

Yuan, Z., X. Zhou, and T. Yang. 2018. Hetero-ConvLSTM: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD '18*, 984–92. ACM Press, London, United Kingdom.

Zhang, G., K. K. W. Yau, and G. Chen. 2013. Risk factors associated with traffic violations and accident severity in china. *Accident Analysis & Prevention* 59:18–25. doi:10.1016/j.aap.2013.05.004.

Zhang, Z., Q. He, J. Gao, and M. Ni. 2018. A deep learning approach for detecting traffic accidents from social media data. *Transportation Research Part C: Emerging Technologies* 86:580–96. doi:10.1016/j.trc.2017.11.027.

Zheng, M., T. Li, R. Zhu, J. Chen, Z. Ma, M. Tang, Z. Cui, and Z. Wang. 2019. Traffic accident's severity prediction: A deep-learning approach-based CNN network. *IEEE Access* 7:39897–910. doi:10.1109/ACCESS.2019.2903319.

Chapter 7

Conclusions and future research

The objectives of this work were two: the development of neural models in a spatio-temporal series prediction context, and the successful application of these models in the field of sustainable mobility and air quality. In general terms, both can be considered as successfully fulfilled. To conclude this work, the highlights of this thesis are summarized below.

Firstly, a new spatio-temporal model based on attention mechanisms was presented. Given that the operation of this model rests on the inclusion of several modules which model different spatiotemporal components, this project has proven to be a perfect opportunity to better understand how neural models manipulate these components. After extensive experimentation over a real traffic dataset, it has been proved that it outperforms some of the most important state-of-the-art neural architectures. Also, its behaviour has been analyzed with respect to both time and spatial dimensions thanks to the interpretable nature of the model. This interpretability layer also has let us illustrate how these kinds of models might be used in order to better understand the problem domain. Concretely, recurrent-based neural networks regulate seasonality and trend, while convolutional-based methods are capable of extracting short-term and spatial relations. Thus, these experiments demonstrate that by properly using each type of neural network, performance can be improved while avoiding redundancies.

Secondly, we have explored how classical spatial assumptions based on closeness are not always the best deal when working with convolutional neural networks for spatio-temporal series regression. This idea has been tested by comparing several versions of convolutional-based models that make no use of prior spatial information with their respective traditional forms through comprehensive experimentation, confirming our main hypothesis: the inclusion of adjacency-based representations of the spatial distribution of real data is not necessarily the best option for the classical convolutional networks. In this work, we also analyzed from a practical point of view how to proceed according to the nature of the data, which will have an important repercussion for the next project.

Thirdly, we have presented SOCAIRE, the new operational system for air quality forecasting and monitoring in the city of Madrid. Built under the idea of combining several statistical and neural models, it shows how it is possible to integrate the diverse information that correlates with air quality in order to model it and boost predictive performance. Specifically, all tools developed in the previous projects converged in this work to generate a model that is capable of bringing together in the most optimal way possible the spatio-temporal information in a real air quality problem. Given its probabilistic nature, the system is also able to combine the predictions of the full probability distribution for compound events which makes SOCAIRE as a highly valuable tool for managing the NO_2 protocol enforced by the city council of Madrid. The tool presented through this project is thus not only a theoretical proposal but it has been adopted as the official application to monitor, analyze and make day-to-day decisions about air quality in the city of Madrid.

Lastly, it is worth mentioning one of the main points presented in Appendix A, which does not specifically follow the line described so far but which, due to its scientific relevance, must be stated here: the lack of reproducibility and experimental rigor in the field of machine learning is a crucial problem, where the ability to expand the field can be undermined by not following an appropriate methodology. This malpractice must be taken into consideration urgently, being addressed by the community unanimously if we intend to build artificial intelligence over a solid and objective foundation.

As a final consideration, we can conclude that we have been able to tackle the wide-ranging frame of reference presented as starting point for this thesis not only in different contexts and problems but also from different perspectives, greatly enriching the presented work. Since science in general, and the context of this work in particular, can be understood as a multidisciplinary prism in which each side allows access to a different interpretative canon, this heterogeneity allows us to conclude this work as a particularly prosperous and productive one.

Future lines of research

Although each project discusses its own future lines, for convenience here is presented a summary of those lines that, once put in context the entire thesis, gather the most important and interesting oportunities. Specifically:

- Attention mechanisms are emerging as one of the most promising methodologies within deep learning due to their capability and versatility in terms of both performance and interpretability. Thus, and although we have demonstrated the actual ability from a spatial-based network to model attention for both input dimensions, space and time, it could be beneficial to extend this idea to outputs dimensions too, having different attention weights for different predicted timesteps.

- From a theoretical point of view, our work is a good starting point to rethink the way of working with spatio-temporal series if we want to extract and make use of the spatial information of the problem more efficiently and beyond the classical adjacency hypothesis. However, this issue requires more pronounced attention if we intend to maximize the extraction of information and knowledge inherent in problems of this nature.

- To work towards the inclusion of a traffic forecasting system for SOCAIRE, which will surely improve the performance of the models by enhancing the information that anthropogenic features provide. Since traffic is known to be one of the main sources of pollution, this could be an important advantage in terms of performance. Moreover, this proposal is directly in line with the current context where the generation of intelligent monitoring systems integrated into municipal services tends to concentrate as many fields of study as possible, thus benefiting from the maximum possible interaction between them.

- Finally, SOCAIRE could be adapted to predict any kind of combined air quality index, and not only those ones affecting the current Madrid protocol. Hence, this framework is easily generalizable in the field of air quality.

Appendix A

Side project II: COVID-19 forecasting with deep learning: a distressing survey

Contribution: Conceptualization, Investigation, Methodology, Validation, Writing.

COVID-19 forecasting with deep learning: a distressing survey

L. Gutiérrez, R. de Medrano and J.L. Aznarte

Abstract—**Building on the success of deep learning techniques in all sorts of classification and regression tasks, in the wake of the COVID-19 pandemic many researchers turned their tools and expertise to the task of predicting the evolution of the infection worldwide. This praiseworthy effort, based on a strong will to help, produced a panoply of models and applications aimed at helping health institutions to plan and decide on the mitigation measures that could control the spread of the pandemic, through forecasting the disease main indicators for public health.**

However, as we show in this paper, this emergency research endeavour has not necessarily been in line with common quality standards in research: it is indeed hard to find papers in which replicability and reproducibility are enabled, lest guaranteed.

After defining a set of quality criteria related to problem definition, dataset management and model identification and evaluation, we studied 96 papers in detail. None of the analysed papers scored positively in all the criteria, while only about one third scored positively in at least half of the defined criteria. These results show that, in the present case, emergency research has been prone to leave behind some of the basic requirements for quality scientific labour.

I. Introduction

Since the World Health Organisation proclaimed the COVID-19 outbreak as a pandemic in March, 2020[1], the spread of the disease has followed certain patterns based on dynamic transmission of the epidemic over time and exhibited a clearly non-linear behaviour. To try to foresee these patterns, during that period, different epidemiological models have been proposed. These models can be split into two wide categories: data-driven statistical models and classical mechanistic models based on epidemiological principles.

The classical epidemiological approach is based on developing compartmental or susceptible–infected–removed (SIR)-like models, which offer a clear epidemiological interpretation. However, predicting with them is sometimes difficult due to strong parameter value ambiguities, mathematical analytic complexity and the assumption that conditions for propagation will remain unchanged [1]. On the other hand, data-driven models use statistical regression practices and machine learning methods to predict how the disease spreads [2]. These machine learning methods are seen as particularly appropriate for predictions based on existing data, being sometimes considered as more accurate compared to common regression models, as they can capture complex and non-linear patterns in the data.

Amongst the most successful ML flavours in recent years, deep learning (henceforth DL) has a prominent place in both scientific and newspaper articles. Despite being sometimes branded as a mere 'buzzword', DL models have been successfully applied to many problems, and are praised frequently amongst the most powerful AI tools. DL comprises complex artificial neural networks with many layers, including models such as deep belief networks, convolutional neural networks, auto-encoders, restricted Boltzmann machines, generative adversarial networks and recurrent neural networks, amongst others.

In the relatively short period since the start of the pandemic, many DL applications for COVID-19 forecasting have been presented, and their performance assessed with a wide variety of metrics. However, forecasting is a challenging and specialized task, especially when dealing with small datasets, and, as in any other scientific discipline, urgencies do not usually favour quality. Predictive models must be carefully evaluated not only on their ability to capture historical events but on their exactitude in forecasting future trends, fostering a stronger appreciation of the technology's capabilities and limitations [3]. Furthermore, this evaluation process must be standardized and then validated by the scientific community.

Notwithstanding, as we will show, most of the available applications of DL to COVID-19 forecasting are affected by common flaws. This worrisome fact raises serious concerns about the maturity of the field, its usefulness in the wake of emergencies and the common publish-or-perish academic career scheme, which has indeed been linked to the so-called replication crisis [4].

Concerning the novelty of our approach, there are indeed previous literature reviews which offer general comparisons of existing machine learning techniques applied to COVID-19 diagnosis and prognosis. To the best of our knowledge, none of them covers the prediction of the spread of the pandemic in an exhaustive manner, and none is focused on DL applications, as shown below.

Therefore, our analysis deals specifically with DL techniques applied to forecasting the number of COVID-19 infected cases, focusing on methodological difficulties and typical challenges that researchers confront. A rigorous quality screening is presented to highlight methodological concerns, emphasizing the weaknesses that can lead to issues about reproducibility and replicability of the results. In order to do so in an objective manner, a set of quality criteria is established beforehand, concerning the datasets used as well as the problem and model definition and evaluation. A set of 96 papers has been studied under these criteria: as we will

jlaznarte@dia.uned.es, corresponding author

All three authors are with the Department of Artificial Intelligence, Universidad Nacional de Educación a Distancia – UNED

[1] https://www.euro.who.int/en/health-topics/health-emergencies/coronavirus-covid-19/news/news/2020/3/who-announces-covid-19-outbreak-a-pandemic

show, the results are all but flattering.

This document is structured as follows: in Section II the existing literature reviews and state-of-the-art papers about the application of DL techniques to forecasting the COVID-19 spread are summarized. In Section III the methodology employed in our work is explained and a set of quality criteria covering several aspects of the scientific process is defined. In Section IV a selection of papers on which DL approaches are applied to COVID-19 time series forecasting are reviewed in light of the aforementioned criteria, aiming at evaluating the replicability and reproducibility possibilities of each one. Additionally, in Section V, we examine the challenges found, and we discuss the most relevant findings. Section VI concludes with a summary of our findings.

II. Related works

Amidst the huge number of papers published since the start of the pandemic in the field of AI applications, a good number of review papers were already available when we decided to initiate our state of the art review. However, not many of them covered forecasting with DL methods (none of them was devoted exclusively to this issue), and thus our research questions were not really answered in the available literature.

When our work started, up to 11 state-of-the-art review papers concerning AI applications to the different aspects of the COVID-19 pandemic were already published [5–16]. Other review papers were released while we actually performed our analysis and were also considered [17–22]. However, most of these works had a broad-spectrum approach, making the target very general and inconclusive, covering any application of AI conceivable and reviewing only very few publications in each line of work. There are also other review papers [23–33], but their primary targets dealt with different applications (i.e. imaging and diagnostics, management, etc.).

For example, in the early work from [5], the fields of study were divided into i) early warnings and alerts, ii) tracking and prediction, iii) data dashboards, iv) diagnosis and prognosis, v) treatments and cures, and vi) social control. However, in that paper, mostly topical and opinion articles released in blogs and newspapers were cited, plus a pair of diagnosis papers and a couple of articles presenting compartmental epidemiological models used for forecasting. In any case, authors devoted little attention to forecasting with DL techniques.

The focus of [6] was divided between blockchain and AI in general, dividing the works into estimation of virus outbreaks sizes, detection and treatment. Within the scope of DL in forecasting, only one paper [34] was mentioned, emphasizing the lack of unified datasets while highlighting the possibility of developing adaptive AI models for predictions. However, this work is too general to answer our questions.

The study carried out by [7] covered the following fields: i) detection and diagnosis, ii) tracking and predicting the outbreak, iii) 'infodemiology' and 'infoveillance', and iv) biomedicine and pharmacotherapy. Authors propose some use cases and mention challenges and solutions, especially the lack of a standard data set, forcing each model to use its own dataset, and making comparisons difficult. They discuss lessons learned and give some recommendations, like the use official datasets from health authorities, the optimizing of algorithms or the integration with other methods. Again, the scope of this work is too wide, since it covers AI in general and all aspects related to the COVID-19 fight. Due to the colossal work that would be to cover all papers in such a wide field, the selection of papers is quite arbitrary. As a result of this, some of them are just mentioned, but none was particularly analysed, resulting in that only two works [34, 35] were related to forecasting cases using DL approaches, both included here.

In [8], the considered categories were i) quick pandemic alert, ii) tracking and diagnosing cases, iii) pharmacological treatment, and iv) public health interventions. A short set of papers were discussed, but only [36] dealt with forecasting with DL, and their final conclusions were very brief and too general.

From the various models analysed in [9], only six were related to DL, while again barely two of them [37, 38] were related to forecasting the spread or cases, and were just briefly commented. Their main conclusion was indeed brief and open: "there is a need of thorough assessment of these predictive analytic algorithm based on type of question to be answered" (sic).

In a more extensive work [10], data sources, classical TS methods, epidemiological models, forecasting, impact and decision-making tools were analysed. In the forecasting chapter, authors mention machine learning, DL, ARIMA and ensemble approaches. They highlighted [39, 40] for fully connected neural networks and [36, 41–43] for recursive neural networks, while approaches dealing with convolutional networks were all devoted to imaging and signal processing. The main conclusions on DL approaches were about the high amount of data required, the complexity of model hyper-parametrization, and the low interpretability of the results. But as the authors themselves admit, their purpose is just to "highlight effective data-driven methodologies that have been shown to be successful in other contexts and that have potential application in the different steps of the proposed roadmap".

In [11], models were divided in four categories: big data, social media/other communication media data, stochastic theory/mathematical models and data science/machine learning techniques. In the latter category, just two papers [37, 38] were relevant to our subject but they were just concisely mentioned. The main challenges they identified were the lack of quality and quantity in data, over-fitted models, overly clean data with eventual integrity loss, data abundance not always improving accuracy, wrong algorithm and attribute selection leading to misleading results, and model complexity that can affect the overall performance. While these questions are important, they are commonly inherent to every data-driven method. Their main conclusion ("it is important to analyse various forecasting models for COVID-19 to empower allied organizations with more appropriate information possible") justifies by itself the existence of our paper. In any case, the variety as well as the number of models that should be analysed must be higher in order to arrive at any sound conclusions.

In another brief paper [12], a few papers were merely enumerated and categorized in i) early detection and diagnosis of the infection ii), monitoring the treatment, iii) contact tracing of the individuals, iv) projection of cases and mortality, v) development of drugs and vaccines, vi) reducing the workload of healthcare workers, and vii) prevention of the disease. From the papers included therein, only [34] was relevant to our subject. With no identified challenge, their conclusions were both wide and general, so very little could be deducted from them.

The divisions in [13] were detection and diagnosis, virology, drug and vaccine development and epidemic. In the latter category, authors dedicated a section to outbreak detection, where a few papers were just described and summarized in a table [34, 36–39, 44–47]. The identified challenges were the lack of large-scale training data and the limited interaction between of computer science and medicine. Still, from this paper it cannot be elucidated which DL methods could be more useful for prediction, or even more, whether DL is useful at all or not.

Deep learning, edge computing and deep transfer learning were the focus of [14]. However, only two of the considered papers [37, 46] were related to our scope. No conclusions could be extracted regarding DL, as its presence was merely testimonial.

In a recent paper [15], only two new citations were added compared to the author's previously mentioned work [5], but they were related to position articles on a blog and a website.

For [16] the main topics were i) screening and treatment, ii) contact tracing, iii) prediction and forecasting, and vi) drugs and vaccination. Only four papers were reviewed for the third category, and only one of them was related to DL techniques. The descriptions and analysis were extensive, including the most important aspects and providing nice explanatory tables. However, the conclusions were brief: "deep learning algorithms [. . .] have more potential, robust, and advance among the other learning algorithms [while] most of the models are not deployed enough to show their real-world operation" (sic). Nevertheless, the only analysed paper within our scope [41] was insufficient to discard a more exhaustive analysis.

In [17] the domains covered were i) detection and diagnosis, ii) contact tracing, iii) forecasting, iv) vaccine development. While this paper is quite exhaustive about the role of AI in computerized tomography (CT) scans and X-Ray images, it only analyses one paper [41] in the forecasting field.

The central subjects for [18] were i) diagnosis using radiography images, ii) diagnosis using respiratory and coughing wave data, iii) severity and survival-mortality assessment, iv) outbreak forecasting models, v) virion sequence formation and drug discovery models. In the forecasting area they provided a list of 27 papers, 12 of them related to DL [35–37, 42, 44, 46, 48–53]. Unfortunately, only four of those papers were actually analysed, while the rest were just depicted in a table by their main features. The identified challenges were: model precision and reliability impacted by quickly constructed datasets and their limited real-world implementations. The final conclusions were that the utility of AI in predicting outbreak and forecasting the spread of COVID-19 is patent but further research is needed to identify real-world uses of AI for COVID-19.

The classification chosen by another exhaustive paper [19], was i) diagnosis, ii) treatment and vaccines, iii) epidemiology, iv) patient outcome and iv) infodemiology. Authors considered 82 studies out of the 435 retrieved, from which only a few [34, 35, 37, 38, 44, 45, 50, 54] were related to forecasting with DL. They analysed the most interesting aspects of the models, like employed techniques, features of the datasets, applications, and publishing countries. Unfortunately, the models were simply summarized in tables. Authors found that papers reported AI features and results inconsistently: for example, approximately one third of them did not disclose the type of validation or the data size, and a few of them did not even specify the type of AI used, thus hampering replicability.

In [20] the considered areas were i) clinical applications, ii) CT and X-ray image processing, iii) epidemiology, iv) pharmaceutical, v) text processing, vi) understanding the virus, and vii) dataset collection. It is in the epidemiology section where we find an exhaustive collection of papers related to forecasting [39, 55–80]. However, those were just described without any further analysis or criticism. Their main conclusions in our field of study were regarding the size of the data, the way they are collected and the variability of formats of these data, while authors propose global search algorithms for training the networks in order to avoid local optima. While those remarks were complete and sharp, they were given from a quite broad perspective.

In [21], the considered applications were protein and drug development, diagnosis and outcome predictions, epidemiology and 'infodemiology'. In the latter category, we can find some modelling and forecasting papers such as [38, 44, 45, 54]. Authors found that "very few of the reviewed systems have operational maturity" and identified three main issues: the need of open global repositories, the creation of multidisciplinary teams, and the need for open science so that solutions can be shared globally and adapted to other contexts.

By the time of finishing this document, a systematic review of the papers covering image-related DL techniques applied to COVID-19 was released [81]. Since time series forecasting and image recognition are entirely different fields, the purpose of that work might be in a similar line to the conclusions extracted here, but there is no overlap.

Summarizing, most of the analysed review papers focus on all the fields related to the fight against COVID-19, or on the variety of AI disciplines available, but, in particular, none is precisely focused on forecasting with DL. As we have seen, the main trend is to describe the methods employed and highlight the overall challenges in a general manner. The lower number of forecasting methods analysed, as well as the predominance of compartmental models, traditional statistical techniques, and conventional machine learning methods versus DL ones, adds up enough evidence to justify the existence of this document.

III. Methodology

A. Paper Selection

As stated above, we focus on DL forecasting approaches related to the prediction of the COVID-19 pandemic outbreak.

Thus, this review focus on works that are using artificial neural networks and more precisely DL techniques to forecast the spread of the COVID-19 pandemic.

According to the European Centre for Disease Prevention and Control (ECDC) [82], the most accurate indicators of epidemic intensity are the absolute number of newly confirmed cases and their notification rate per 100,000 population. Hence the output of the considered models must be, at least but not limited to, the number of newly confirmed cases. This indicator is usually complemented with the number of total cases, active cases, recovered cases, deceases, and other measures. On the other hand, the inputs will usually be the number of total recorded (confirmed) cases, but they may be accompanied by the recorded number of total cases, active cases, recovered cases, deceases etc.

For the sake of simplicity and standardisation, the models proposed in the reviewed papers were sorted amongst one of the following categories:

- Artificial neural networks (ANN) [83, 84]: multilayer perceptron [85] (MLP) or feed-forward multilayer neural network (FFNN) [86], Autoregressive Networks [87], Auto-encoders [88, 89], Adaptive Networks [90].
- Recurrent Neural Networks (RNN) [91]: Long Short-Term Memory units (LSTM) [92], Gated Recurrent Units (GRU) [93], Bidirectional RNNs (BRNN) [94], Multi-head attention (ATT) [95].
- Convolutional Neural Networks (CNN) [96].
- Extreme learning machines (ELM) [97].
- Ensemble methods.

Other denominations, such as Deep Neural Networks (DNN) [98], could have been ascertained into any of the previous categories, being the 'deep' characteristic an arbitrary boundary.

We consider studies published in English between 1 January 2020 and 10 May 2021, including conference proceedings, dissertations, peer-reviewed articles, and preprints. Any other publications such as blogs, topical papers, opinion essays or commentaries, were discarded. We did not contemplate any limitations regarding the origin of publication, study design, or outcomes. Out of the several hundred titles retrieved through a systematic search and independent screening by titles and abstracts, 97 studies were retained for full text reading. The selected ones were crosschecked with the cited bibliography of the reviews already discussed in the previous sections, resulting in the addition of a few more papers to our study.

The search was performed in well-known databases like ResearchGate, SpringerLink, Elsevier, IEEE Xplore, ACM Digital Library, arXiv, medRxiv, and Google Scholar, excluding terms like 'sentiment', 'drug', 'X-Ray', 'Computer Tomography', 'Imaging', 'RNA' etc. or any of its variants. For an example of the queries used, see Figure 1.

B. Assessment Criteria

In order to assess the quality of every considered paper, following the lead of previous meta-analysis as explained in Section II, in order to make our analysis as fair as possible, we need to define a set of criteria. These criteria or key quality indicators must represent concrete, measurable features of the papers, and must be as objective as possible. In this section, the set of key quality indicators that have been chosen for comparison of the selected papers is described. These indicators aim to assess the information that quality papers must provide to the reader, in order to evidence the robustness of the model, to elucidate the conditions of the study, to explain how uncertainty is managed, and to guarantee future replicability.

In relation to concerns expressed in previous works about how AI, ML or DL are applied in the field of medicine [99–104], our work is rooted in existing paper evaluation frameworks [105, 106], which we have adapted to the specific needs of the chosen field. Despite the sharp and useful recommendations from [104], it is mainly focused on clinical trials, and thus its main purpose is to be a guideline for developing studies rather than a literature review. From the list of items described in [105], while some of them are common to any kind of AI study, and hence applicable to our problem, the majority is exclusively applicable to medical imaging. Therefore, while the medical imaging items were not considered here, the general principles were assumed in order to elaborate our list of criteria. Finally, specific criteria related to forecasting were also added to the list.

Below we describe the set of considered criteria, which are classified according to their focus.

1) Criteria related with the problem description: In any case, to be considered as a quality paper, any article must include a specific and clear description of the problem to be solved, stating the dependent and independent variables that are considered, the area of study, the forecasting horizon, the period of study, and the employed techniques (i.e., type of ANN). Authors should avoid ambiguous assertions like 'predicting the curve', 'forecasting the spread', 'foresee the evolution', etc., favouring clear statements about measurable variables.

1) Object of study. The paper must clearly indicate what is the goal of the study, the type of predictive modelling to be performed, the target variables to be predicted, and the characteristics of the variables which are inherent to the problem description and have a direct effect on the replicability of the experiment: area of study (province, state, region, country), variables to predict (cases, deaths, recoveries), etc.
2) Model identification. The chosen forecasting models must be properly identified and presented, citing previous works in case the models are not new.
3) Forecast horizon. The study must specify the time lag into the future for which forecasts are to be prepared. In the COVID-19 forecasting case, this will vary from short-term forecasting horizons (weeks) to long-term horizons (years) [107]. The chosen forecasting horizon may have a direct impact on the prediction error [108] as well as on the usability of the results.

2) Criteria related with the datasets: Any good paper in this context must contain a clear description of the dataset and the data curation procedures applied, including availability and any transformations in the ETL process. This is especially

Search examples:

COVID forecasting "deep learning" -images -drug -sentiment -RNA -symptoms -X-Ray -CT -Radiograph
COVID forecasting Convolutional -images -drug -sentiment -RNA -symptoms -X-Ray -CT -Radiograph
COVID forecasting LSTM -images -drug -sentiment -RNA -symptoms -X-Ray -CT -Radiograph
COVID forecasting Recurrent -images -drug -sentiment -RNA -symptoms -X-Ray -CT -Radiograph
COVID forecasting RNN -images -drug -sentiment -RNA -symptoms -X-Ray -CT -Radiograph
COVID forecasting CNN -images -drug -sentiment -RNA -symptoms -X-Ray -CT -Radiograph

Figure 1: Example of some search constraints employed.

important in the COVID-19 forecasting framework since the data are far from consolidated.

4) Data sources. The paper must clearly state the sources of the data, providing links to them and/or depositing the data tables used for modelling in a publicly accessible repository.
5) Features. Variables contained in the dataset (cases, deaths, recoveries, etc.) and the area where the data is circumscribed to (province, state, region, country, hospital) must be properly described in the document.
6) Study interval. The paper must explicitly include the initial and final date for the considered dataset, providing a clear view of the dataset size and the period analysed.
7) Missing data handling. The paper must specify how inconsistent, missing and/or wrong data points were handled.
8) Data preprocessing. How raw data from various sources was converted into a time series must be clearly specified, as well as any use of normalization, rescaling and/or standardization.

3) Criteria related with the model description:

9) Software. The paper must specify the names, version numbers and configuration settings used in any software, libraries, frameworks and packages used in the experiments.
10) Accessibility. The paper must state a publicly accessible repository where the full code of the modelling process can be found, in order to allow replication and a better interpretation of the study.
11) Initialization. The paper must indicate how the initial parameters of the models were fixed, specifying the distribution from which random values were drawn for any randomly initialized parameters, as well as any random seeds necessary. If transfer learning is employed, the source of the starting weights must be clarified, or the weights provided. When there is a combination of random initialization and transfer learning, it must be clear which portions of the model were initialized with which strategies.
12) Topology. The number of layers and how they are connected must be clearly and fully specified in the paper.
13) Activation functions. The paper must specify the number and type of cells on every layer, and the type of activation function selected in every one of them [109].
14) Objective function and optimizer. The paper must precisely describe the function to be optimized, also called the cost function, loss function, or error function in minimization problems [110], as well as the chosen optimizer and how it has been parametrized [111].

4) Criteria related with the model evaluation: Cross-validation and bootstrapping are validation methods that are typically used for the evaluation of model performance or for fine-tuning. Alternatively, hold-out validation may address the internal validity of a model but would not accurately assess its generalizability [99]. Moreover, using hold-out in small datasets may lead to biased predictions, and in that case results will be dependent on how the data is split into train and test sets. Cross-validation can provide a better indication of how well the model will perform on unseen data, as it gives the opportunity to train on multiple train-test splits [112]. An honest validation procedure should reveal the optimism that is associated with the full modelling procedure, since model uncertainty usually is more important for optimism in model performance than parameter uncertainty [113].

Despite statistical testing for calibration is not without pitfalls [114–116], when p-values are reported with sensible precision (i.e., $p = 0.023$, instead of the conventional $p < 0.05$), together with 95% confidence intervals, the consistency between the results obtained and pure chance can be measured, thus providing a better understanding of the results.

15) Validation. The papers must clearly specify how the results were validated (hold-out, cross-validation, rolling validation, etc.) and how data were assigned into training, validation, and testing partitions.
16) Error metrics. The papers must clearly describe the error metrics employed to assess the model's performance and choose appropriate and well-known metrics for forecasting problems [117].
17) Benchmark comparison. The performance of the AI model must be compared against state-of-the-art models and naïve models.
18) Statistical inference. The papers must state what kind of hypothesis tests have been applied in order to decide whether experimental results contain enough information to cast doubt on conventional wisdom.

5) Final score:

19) Final score. Meant as a summary of the set of criteria described above, this score will be computed as the sum of the number of criteria that each paper meets completely and explitly. Only in case of a draw, we will recourse to comparing the number of criteria that are met in an implicit way (see † in the following section), and then those which are just partially met (see ‡below).

Table I: Summary of scores per field: N (no), Y†‡(implicit and partially yes), Y‡(partially yes), Y† (implicitly yes), Y (yes).

	N	Y†‡	Y†	Y‡	Y
Object of Study	0	1	30	5	**60**
Forecast horizon	23	0	0	0	**73**
Data Sources	5	0	0	0	**91**
Features	4	1	14	11	**66**
Dataset Interval	13	0	0	1	**82**
Missing data handling	**80**	0	2	0	14
Data Pre-Processing	**45**	0	0	8	43
Software	**43**	3	1	31	18
Accessibility	**88**	0	0	0	8
Initialization	**76**	0	4	8	8
Topology	19	0	0	1	**76**
Activation Functions	28	0	0	29	**39**
Objective Function & Optimizer	25	0	0	33	**38**
Validation	26	0	0	**57**	13
Error Metrics	14	0	0	0	**82**
Benchmark Comparison	18	0	0	**45**	33
Statistical Inference	**88**	0	0	0	8

IV. Analysed models

At the time of writing, several papers about DL applications to COVID-19 have been retracted [118], in yet another hint to worrisome flaws in the quality of science in emergency times. However, none of them dealt with forecasting except one, which was indeed withdrawn on 10 Nov., 2020 [119], leaving the total amount of considered papers in 96.

All those works were evaluated against each of the criteria defined above. Papers were marked with an "N" when they did not meet the criterion, and with a "Y" when it was fully satisfied. Papers were awarded a "Y with reservations", when criteria were partially met, for example in cases when the required information could be only found implicitly throughout the text (†), or only partially (‡) or both (†‡).

A. Problem Description

To stress the potential novelty of their models, certain authors tend to give imaginative or elaborate names to them, sometimes difficulting the identification. Nevertheless, all the models found in the considered papers were classified according to the model taxonomy detailed in **Section III-A**. Amongst the 95 analysed papers, a total amount of 143 models were employed. The most popular model was LSTM, followed by FFN, while GRU and CNN ranked in 3rd and 4th position (see **Figure 2**).

All the considered papers explicitly state the object of study, albeit with different fortune. For example, 12 of them [37, 41, 47, 48, 59, 62, 69, 120–124] did this only in an implicit way, by distributing the information throughout the text. Only [38] and [125] did this in a partial manner, not mentioning the variables to predict. The information provided by the former was implicit.

Surprisingly, from all the analysed papers, 23 of them did not explicitly state the forecast horizon employed [44, 52–56, 60, 69, 70, 73, 80, 120–122, 125–133], while the rest were found to do so in one way or another.

B. Data

From all the reviewed papers, only 5 failed to state the source of the datasets used [54, 77, 120, 134, 135]. The other

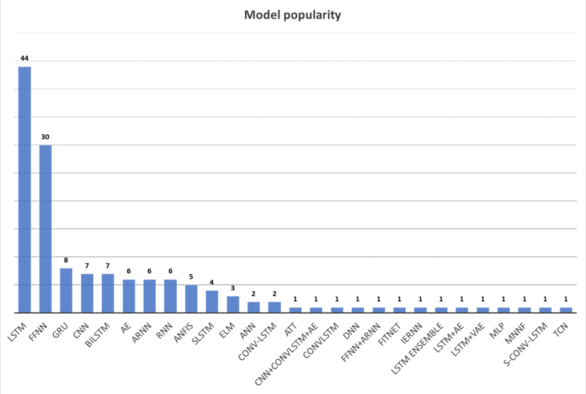


Figure 2: Number of times each type of models has been proposed in the set of considered papers.

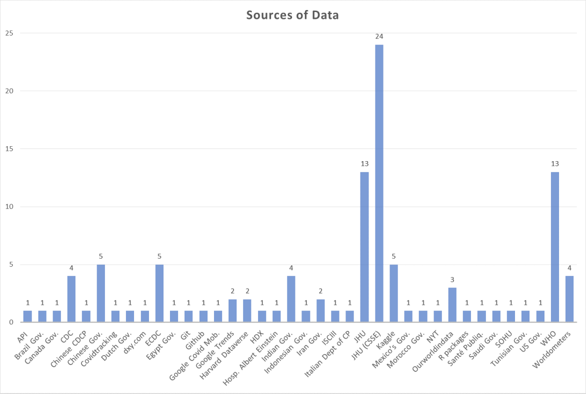


Figure 3: Number of times each source of data has been used in the considered papers.

89 mentioned up to 110 data sources in total. As can be seen in Figure 3, 21 of those employed governmental data, while twelve more used data from local or regional health agencies, such as Centers for Disease Control and Prevention (CDC), European Center for Disease Control and Prevention (ECDC), Chinese Center for Disease Control and Prevention CCDC, etc. The most popular data source was the repository of the Johns Hopkins University (JHU), mentioned 37 times, whereas The Center for Systems Science and Engineering (CSSE) at JHU was specifically mentioned in only 24 of them. The main international organisation mentioned was World Health Organisation (WHO) (30 times), while publicly accessible data repositories were relatively popular: Kaggle was mentioned 5 times, Worldometers 4 times and OurWorldInData 3 times. Only two private repositories were found to be considered: an API with authorised access from [79] and hospital data from [80]

When describing the features present in the dataset, the results are more heterogeneous: 4 papers failed to report any detail at all [77, 133, 134, 136], while 10 of them only described the features partially [39, 41, 50, 52, 54, 57, 126, 131, 137, 138], while [139] did it only in an implicit manner. Other 14 provided this information implicitly and distributed

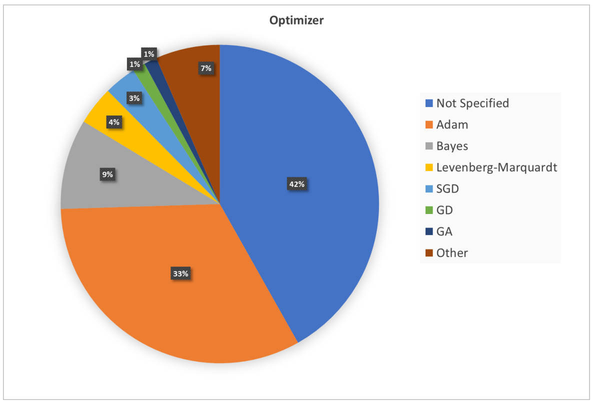


Figure 4: Number of times each optimizer has been used in the set of considered papers.

throughout the whole document [42, 53, 59, 61, 63, 64, 75, 122, 124, 132, 140–143]. The most common reason for this is that the variables are not specified (new cases, accumulated deaths, etc.) or even the area where the data belongs to is not declared.

The considered time interval (and thus the length of the dataset) was not stated in up to 13 of the analysed papers [58, 61, 72, 77, 121, 128, 130, 132, 133, 136, 137, 139, 144]. As can be seen in Appendix **??**, this size varies from only 14 days used by [38] up to two hundred and eighty-four days from [145]. The average size of the dataset was 100.36 days with a standard deviation of 56.19.

With respect to missing data, only 14 studies stated how they dealt with this problem [43, 53, 61, 68, 71, 123, 130, 140, 146–151], while 2 others did this implicitly [22, 135]. The rest of the papers did not mention anything about this aspect, which does not mean that they failed to approach the issue. The lack of missing data might be behind this, but it is always a good practice to explicitly state it. Amongst the papers which dealt with missing data, the approaches are heterogeneous. For example, missing data was just left blank [53], simply eliminated [71, 130, 146], or no missing data found [150]. Others replaced the missing data by the average of five previous and posterior data points [61], or by an average of one week of data [140] or by reversed order values from the sequence [123] or by using linear weighted moving average [149].

Up to 45 of the papers did not mention anything regarding what kind of data pre-processing was applied [35, 39, 42, 44, 46, 49–52, 54–57, 61–65, 67, 69, 71, 77, 80, 121, 124, 130, 132–135, 138, 142, 144–146, 148, 149, 152–158], while 8 only acknowledged this partially [53, 72, 75, 122, 123, 143, 159, 160]. While this does not necessarily mean that data was not pre-processed, these kind of inscrutabilities obviously hinder replicability. The most widespread practice was minimax normalization.

C. Model description

Only 18 works fully documented the software packages and libraries employed, including the versions [37–39, 46, 52, 55, 73, 74, 77, 128, 134, 142, 151, 154, 156, 159, 161, 162]. While [80] did not make it explicit, it was possible to infer it from the source code. Only the name of the software could be implicitly extracted from the repository in [62, 72, 155], which is not a recommended practice. From the rest, 31 papers only revealed the software name [22, 40, 42, 47, 48, 58, 61, 64, 68–71, 75, 79, 122–125, 130–133, 147–149, 157, 160, 163–166], while the others did not include any mention at all. This practice leads to difficulties in reproducibility.

Only 8 papers decided to provide a repository where the full experimental protocol could be accessed [48, 55, 62, 80, 120, 155, 156, 162]. This is an opaque practice that does not favour replicability.

Concerning the initialization of the model, 8 of the articles undisclosed the chosen way for initializing the weights in an unambiguous manner [38, 62, 79, 123, 128, 131, 134, 151], while 4 more mentioned this just in an implicit way [55, 80, 155, 162]. Other 8 decided to provide this information just in a partial way, only declaring that this was done randomly, but without specifying which distribution was used [22, 43, 51, 63, 120, 133, 150, 156]. The rest did not make any mention at all about this aspect.

Regarding network topology, up to 19 of the works failed to mention the number and type of layers from which the network was made up [38, 41, 42, 56, 57, 64, 67, 121, 128, 130, 133, 138, 140, 142, 148, 152, 153, 155, 164]. Only [120] partially did this task, while the rest clearly stated this fact.

From all of the analysed papers, up to 27 failed to explain the number of units employed in each layer and their activation functions [13, 38, 41, 42, 45, 54, 56, 57, 60, 61, 64, 69, 77, 121, 124, 128, 130, 133, 140, 142, 148, 152–155, 161, 164]. Other 29 did it only in a partial way [34, 36, 37, 39, 44, 46, 47, 49, 52, 53, 62, 63, 74, 76, 79, 120, 122, 125, 126, 131, 132, 136, 138, 144, 156, 157, 159, 162, 166], while the rest made this information explicit and complete. The most popular was ReL followed by Sigmoid + Tanh (due to the popularity of LSTMs) and standalone Tanh.

Up to 25 of the works failed to describe the selected objective function, and/or the optimizer applied to minimise it [42, 44, 52–54, 57, 61, 63, 67, 70, 74, 78, 128–130, 134, 138, 140, 142–144, 154–156, 166]. Another 32 succeeded in this task only partially [37, 38, 41, 45–49, 55, 56, 59, 62, 64, 69, 71, 73, 76, 120–125, 127, 131, 136, 139, 145, 148, 152, 153, 164]. As can be seen in Figure 4, the most frequently chosen optimizer was Adam, followed by far by Bayesian optimizer.

D. Evaluation

Concerning evaluation, only 13 of the analized papers informed about a full cross-validation method [37, 43, 46, 55, 63, 71, 80, 124, 125, 140, 151, 159, 166], while other 25 did not mention if any kind of validation was performed at all [34–36, 38, 47, 53, 54, 57–59, 61, 72, 77, 121, 132, 133, 137, 138, 142, 144, 146–148, 158, 165]. This worrisome fact undoubtedly makes the interpretation of the results an exercise of faith. The 57 remaining papers only performed a 1-split hold-out validation, which of course can introduce some bias in their conclusions, especially when the size of the

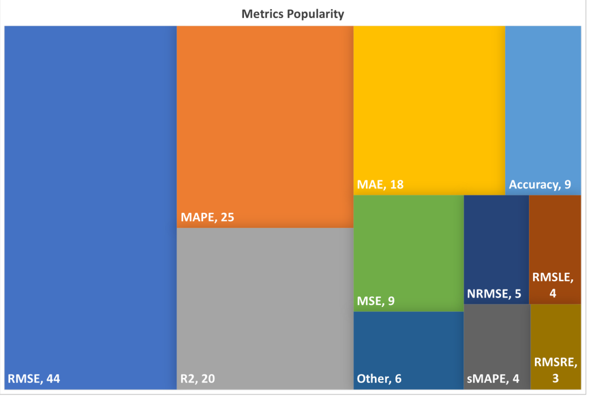


Figure 5: Number of times each error metric has been used in the set of considered papers.

dataset is small. The most common split rate was 80/20 for test and validation, respectively, followed by 90/10 and 75/25. Only [153, 156] applied a 50/50 split, and [73] even went for 40/60. This practice is not advisable, especially with such small datasets, as models will find more difficulties to learn the general principles and will show a poor validation and test performance.

Unfortunately, up to 14 papers failed to provide any error metric at all [36, 50, 56, 57, 62, 64, 67, 77, 121, 122, 126, 129, 142, 144]. Furthermore, while the problem at hand is clearly a regression one, 8 of the studies employed only metrics for classification, making difficult to understand how far the predictions were from the actual values [80, 136, 139–141, 148, 154, 157]. Particularly, [136, 139] used an own formula in an effort to 'adapt' accuracy metric to prediction problems. Another five articles used a mix of classification and regression metrics, leaving some room for comparisons [41, 58, 125, 128, 145]. The rest provided only regression metrics. From Figure 5 it is clear that the most common metric was RMSE, followed by MAPE, R^2 and MAE, while up to 6% of the times, accuracy was chosen.

As mentioned above, comparison with naive and state-of-the-art models is key to prove the goodness of any forecast attempt. Of all the analized papers, 18 papers did not include any kind of benchmark comparison against any other model [34, 41, 42, 51, 53, 55–57, 62, 66, 77, 129, 130, 134, 141, 158, 163, 164]. Another 31 only compared their proposals against complex algorithms, assuming that all of them are thus better than basic persistence or random approaches, which may lead to problematic conclusions [22, 37, 38, 46, 47, 59, 61, 63, 64, 67, 70, 75, 76, 78, 126–128, 131, 132, 136, 138, 139, 142, 143, 145, 151–155, 157].

Less than 10% of the papers reported the application of some kind of statistical inference to their results, and thus, for the rest, it is difficult to assess that the true gain of the model is not due to simply chance [22, 46, 65, 76, 127, 133, 140, 155].

Regarding confidence intervals, only 18 papers [22, 36, 38, 41, 42, 57, 62, 65, 67, 69, 76, 120, 123, 140, 142, 144, 145, 152] employed them to communicate their results, while [156] mentioned this during the training phase only. From those ones mentioned, solely a few of them [57, 65, 123, 144] employed the intervals for accompanying the numerical results, while the rest only applied them for the charts. As an example, [38, 42] only used them for only one out of the several curves provided.

In particular, only 11 of those [36, 42, 62, 65, 76, 123, 140, 142, 144, 145, 152] provided a 95% confidence interval, while [22, 57] employed a threshold of 80% for their uncertainty intervals, but not in the article, but in a website that supports their paper.

Some singular practices were found, for example in [148], where predictions were made with ±50% of the predicted value, and some of the charts depicted an interval whose level of confidence was not defined. On the other hand, [132] provided the metric values with their mean and their variance, which at least provide some additional information about the fitness of the model. In an attempt to capture uncertainty, in [136, 139] metrics were delivered for different error margins (from 0.05 to 0.5, in steps of 0.05).

Finally, [38, 41, 42, 67, 69, 120] did not mention any numerical indication for the confidence threshold. This practice in particular, together with the absence of any confidence intervals at all, makes more difficult to interpret the uncertainty in their predictions, as the estimated probability of capturing the truth is ambiguous. The rest did not employ any kind of confidence interval, or at least, failed to mention it.

E. Final score

After applying all the criteria, only a maigre 35 of the 96 studied papers scores in *at least half* of the fields. The best score overall was 15 over 17, obtained by [151], only failing in the statistical inference and the accessibility fields, as can be seen in Figure 6. It is followed by [22, 43], both with a positive score in 12 of the fields, and both failing in accessibility. But [22] provides information about how missing data was handled in an implicit way, while [43] totally fails in managing statistical inference and in providing information about the software.

On the other side of the ranking, [121] scores only in the data source and features description, while the object of study is only available implicitly throughout the whole document.

V. Discussion and Recommendations

The outstanding efforts to model and forecast the COVID-19 pandemic using deep learning techniques are obvious and should be praised. Nonetheless, the predominance of methodological and reporting insufficiencies has been also patent throughout our analysis. In fact, none of the papers fulfills all of the proposed quality criteria. Remarkably, [22] was the only one failing in a single criterion, followed by [123, 151, 162] which failed in only two of them. The paper fulfilling more of the criteria without any reservations is [151].

As can be seen in Figure 7, the most common weaknesses are related to the lack of application of statistical inference (in 87 articles), poor definition of the experiments (again

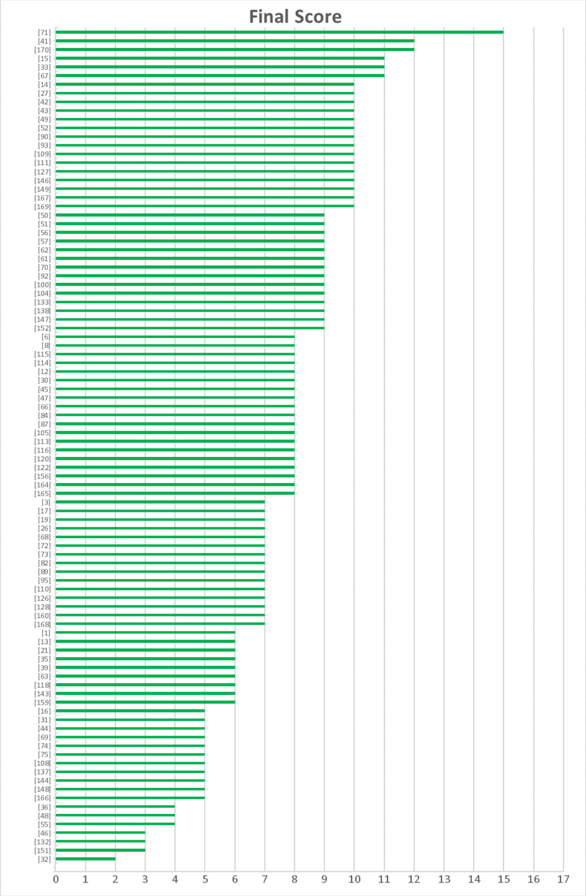


Figure 6: Final scores recorded for the considered papers.

in 87 of the papers), missing data handling (in 79 of the works), missing model initialization details (in 76 papers), no information on data pre-processing (in 44 of them) and lack of software information (in 43 of the articles). These issues may lead to excessively enthusiastic performance estimations and reduced replicability.

When dealing with criteria related to the problem definition, our findings reveal that sometimes it is not clear what the target of a paper is due to the use of ambiguous terms or incomplete assertions, such as "predict COVID-19 infection" or "forecast COVID-19 outbreak". The enunciation of a forecasting problem, in opposition to other types of AI endeavours, should not be necessarily a difficult task. It should be enough to explicitly state the goal of the study, the target variables (i.e., "number of COVID-19 confirmed cases"), the region the predictions are being made for (i.e., "China", "Emilia-Romagna" or "Hospital Albert Einstein"), the forecasting horizon (i.e., "in the next ten days") and the model employed (i.e., "a stacked LSTM model"). So, simple and clear statements explicitly establishing these factors should be a requirement for any paper describing such an application.

Concerning the data-related criteria, the small volumes of COVID-19 data, the different dataset sizes, the diverse kind of variables collected within the datasets, and the way this data is collected by the different organizations and governments remains a huge challenge for an accurate model comparison. We agree with [7] in suggesting that the use of big collaboratively and high-quality datasets provided by governments and healthcare organizations (i.e., WHO, CDC, JHU, etc.) may help to overcome this issue. The surveillance on the quality of the aggregated data by renowned organizations can help to avoid 'retrodden' datasets and may reduce over-fitting, derived from the fact that the community is focused on outperforming benchmarks on a single public dataset.

However, in our opinion, there are no excuses for not explicitly stating the data sources, for example, as well as a clear description of the variables and the intervals considered, or the decisions taken about missing data or the pre-processing stages.

With respect to model description, in a research environment in which open science is becoming more and more encouraged, and for the sake of interpretability and replicability, it is common sense to reveal as much details from the model as possible, so the experiments can be reproduced, and models can be compared to future research.

The disclosure of the software packages, frameworks, and libraries employed, as well as its versions, can certainly enhance the understanding of the performance and conclusions derived from any experiment, while enabling replicability.

Similarly, when dealing with neural networks, revealing the number of layers, number of units, and the activation functions, together with the objective and optimizer function, becomes essential to understand the developed model and its eventual advantages and drawbacks.

Another potential source of obscurity is the randomization of the weights [137], being one of the main sources of stochasticity of the model. Unveiling the distribution from which the weights are being initialized, as well as the employed seeds, is crucial in enabling the reproducibility of any investigation in this field.

Finally, access to the source code and the original dataset employed enhances the comprehension of the model itself and eases the endeavour of repeating someone else's experiments. In such sense, making the complete experiment framework available in a public repository is a practice that boosts the progress of science, especially in these challenging times.

Regarding the evaluation of the proposals, there is no unique appropriate metric for model errors. Using RMSE leads to large errors having a relatively greater influence on the total compared to the smaller ones [167]. This makes MAE better for discriminating among models. Despite its robustness against outliers, MAE is more sensitive to variance, fluctuating its value between several errors sets with the same RMSE [168].

RMSE might be selected to minimize cost function because it helps to calculate the gradient of absolute errors. It is known that with a low number of samples (i.e. 100), giving the values of the errors themselves is probably better than any statistics. Otherwise, large outliers might be excluded from the RMSE

Ref	1	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Final Score
[1]	Y†	Y	Y	Y†	Y	N	N	Y‡	N	N	Y	N	Y‡	Y	Y	Y‡	N	6
[3]	Y†	Y	Y	Y	Y	N	N	Y‡	N	N	Y	Y‡	Y	Y‡	Y‡	Y	N	7
[6]	Y‡	N	Y	Y	Y	N	N	Y†	Y	Y†	Y	Y	Y	Y	Y‡	Y‡	N	8
[8]	Y†	N	Y	Y	Y	N	Y	Y‡	N	N	Y	Y	N	Y‡	Y	Y	N	8
[12]	Y	N	Y	Y‡	Y	N	Y	Y‡	N	Y	Y	Y‡	Y‡	Y‡	Y	Y	N	8
[13]	Y‡	Y	Y	Y‡	Y‡	N	Y	N	N	N	Y	N	Y‡	N	Y	Y	N	6
[14]	Y	Y	Y	Y	Y	Y	N	Y‡	N	N	Y	Y	Y	Y‡	Y	Y‡	N	10
[15]	Y	Y	Y	Y	Y	N	N	Y	N	N	Y	Y‡	Y‡	Y	Y	Y	Y	11
[16]	Y	Y	Y	Y†	Y	N	N	Y‡	N	N	N	N	Y‡	Y‡	N	Y	N	5
[17]	Y	Y	Y	Y	Y	N	Y	Y‡	N	N	N	N	Y‡	Y‡	Y	N	N	7
[19]	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	Y‡	Y‡	Y	Y	N	7
[21]	Y‡	N	Y	Y	Y	N	Y	Y‡	N	N	Y	Y‡	Y‡	Y	Y†	Y‡	N	6
[26]	Y†	Y	Y	Y	Y	N	N	Y†‡	Y	Y	Y	Y‡	Y‡	Y‡	N	N	N	7
[27]	Y	N	Y	Y	Y	N	N	Y	Y	Y†	Y	Y	Y‡	Y	Y	N	N	10
[30]	Y	Y	N	N	Y	N	N	Y	N	Y	Y	Y	N	Y‡	Y	N	N	8
[31]	Y†	N	Y	Y†	N	N	N	Y‡	N	N	Y	Y‡	Y	N	Y	Y	N	5
[32]	Y†	N	Y	Y	N	N	N	N	N	N	N	N	Y‡	N	N	Y‡	N	2
[33]	Y	Y	Y	Y	Y	Y	Y	Y‡	N	N	Y	Y	Y	Y‡	Y	Y‡	N	11
[35]	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	Y‡	N	Y	N	6
[36]	Y†	Y	Y	Y‡	Y	N	Y	N	N	N	N	N	Y‡	Y‡	Y†	N	N	4
[39]	Y†	Y	Y	Y‡	Y	N	N	N	N	N	Y	Y	Y	Y‡	N	Y‡	N	6
[41]	Y	Y	Y	Y	Y	Y†	Y	Y‡	N	Y‡	Y	Y	Y	Y‡	Y	Y	Y	12
[42]	Y	Y	Y	Y	Y	Y	N	N	N	N	Y	Y	Y	N	Y	Y‡	N	10
[43]	Y	Y	Y	Y	Y	N	Y	N	N	N	Y	Y	Y	Y‡	Y	N	N	10
[44]	Y	Y	Y	Y	N	N	N	N	N	N	Y	Y‡	N	N	N	Y‡	N	5
[45]	Y†	Y	Y	Y	N	N	Y	Y‡	N	N	Y	Y	Y	N	Y	Y‡	N	8
[46]	Y	N	N	Y‡	Y	N	N	N	N	N	Y	N	N	N	Y†	Y‡	N	3
[47]	Y	Y	Y	Y	Y	N	N	N	N	N	Y	Y	Y‡	Y‡	Y†	Y	N	8
[48]	Y	N	Y	N	N	N	N	Y‡	N	Y‡	N	N	Y	N	Y	Y	N	4
[49]	Y†	Y	Y	Y	Y	N	Y	Y	N	N	Y	Y‡	Y‡	Y	Y	Y	N	10
[50]	Y†‡	Y	Y	Y	Y	N	Y	Y	N	Y	N	N	Y‡	N	Y	Y	N	9
[51]	Y†	Y	Y	Y	Y	N	Y	Y‡	N	N	Y	Y	Y	N	Y	Y‡	N	9
[52]	Y	Y	Y	Y	Y	N	Y	Y‡	N	N	Y	Y	Y	Y‡	Y	N	N	10
[55]	Y	N	Y	Y	Y	N	N	N	N	N	N	N	Y‡	Y‡	N	N	N	4
[56]	Y‡	Y	Y	Y	Y	N	Y	N	N	N	Y	Y	N	Y‡	Y	Y	N	9
[57]	Y	Y	Y	Y	Y	N	N	Y	Y	Y‡	Y	Y‡	N	Y‡	Y	Y‡	N	9
[61]	Y	Y	Y	Y	Y	N	Y	N	N	N	Y	Y‡	Y	N	Y	N	N	9
[62]	Y	Y	Y	Y	Y	N	N	N	N	N	Y	Y	Y	N	Y	Y‡	N	9
[63]	Y	N	Y	Y	Y	N	N	N	N	N	Y	Y‡	N	Y‡	Y	Y‡	N	6
[66]	Y‡	Y	Y	Y	Y	N	N	N	N	Y‡	Y	Y	Y	Y‡	Y	N	N	8
[67]	Y	Y	Y	Y	Y	Y	Y	N	N	Y‡	Y	Y	Y	Y‡	Y	Y‡	N	11
[68]	Y†	Y	Y	Y†	Y	N	N	N	N	Y‡	Y	Y‡	N	Y	Y	Y	N	7
[69]	Y†	N	Y	Y‡	Y	N	N	Y	N	N	Y	Y‡	N	Y‡	Y	Y‡	N	5
[70]	Y	Y	Y	Y†	Y	N	Y‡	Y‡	N	N	Y	Y	Y	Y‡	Y	Y	N	9
[71]	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y‡	Y	Y	15
[72]	Y	Y	Y	Y†	N	Y	N	Y‡	N	N	Y	N	N	N	Y	Y	N	7
[73]	Y†	Y	Y	Y†	Y	N	Y‡	N	N	N	Y	Y	N	Y‡	Y	Y	N	7
[74]	Y†	Y	Y	Y‡	Y	N	N	N	N	N	N	Y‡	N	N	Y	Y	N	5
[75]	Y†	N	Y	Y†	Y	Y	Y‡	N	N	N	Y	Y‡	N	N	Y	N	N	5
[82]	Y†	Y	N	Y	Y	Y†	N	N	N	N	Y	Y	Y	Y‡	Y	Y‡	N	7
[84]	Y	Y	Y	Y	Y	N	N	Y	N	N	Y	N	N	Y‡	Y‡	Y	N	8
[87]	Y	Y	Y	Y	Y	N	N	Y‡	N	N	Y	Y‡	N	Y	Y	Y‡	N	8
[89]	Y†	Y	Y	Y	N	N	Y‡	Y†‡	N	N	Y	Y	Y	N	Y	Y‡	N	7
[90]	Y	Y	Y	Y	Y	N	Y	N	N	N	Y	Y‡	Y‡	Y‡	Y	Y	Y	10
[92]	Y†	Y	Y	Y	Y	Y	Y‡	Y‡	N	Y	Y	Y	Y‡	Y‡	Y	Y‡	N	9
[93]	Y	N	Y	Y	Y	N	Y	N	N	N	Y	Y	Y‡	Y‡	Y	Y	Y	10
[95]	Y	N	Y	Y	Y	N	Y	N	N	N	Y	Y	N	Y‡	N	N	N	7
[100]	Y	Y	Y	Y	Y	N	N	Y†‡	Y	Y†	N	N	N	Y‡	Y	Y	Y	9
[104]	Y†	Y	Y	Y	Y	N	Y	Y‡	Y	N	Y	Y	Y‡	Y‡	Y	Y‡	N	9
[105]	Y	Y	Y	Y‡	N	N	Y	N	N	N	Y	Y	Y	N	Y	Y‡	N	8
[108]	Y†	N	Y	Y	Y	N	N	Y‡	N	N	Y	N	Y‡	Y‡	Y	Y‡	N	5
[109]	Y	Y	Y	Y	Y	Y	N	Y‡	N	N	Y	Y	Y‡	Y	Y	Y‡	N	10
[110]	Y	N	Y	Y‡	Y	N	Y	N	N	N	Y	Y‡	Y	Y‡	N	Y	N	7
[111]	Y†	Y	Y	Y	Y	Y	Y	Y‡	N	N	Y	Y	Y	N	Y	Y‡	N	10
[113]	Y†	Y	Y	Y	Y	N	Y	Y‡	N	N	Y	Y‡	Y‡	N	Y	Y	N	8
[114]	Y	N	Y	Y	Y	N	Y	N	N	N	Y	N	Y	Y‡	Y	Y‡	N	8
[115]	Y	Y	Y	Y	Y	N	Y	N	N	N	Y	N	Y‡	Y‡	Y	Y‡	N	8
[116]	Y	Y	Y	Y†	Y	N	Y	N	N	N	Y	Y	Y	Y‡	Y‡	N	N	8
[118]	Y	Y	Y	Y	Y	Y	N	Y‡	N	N	N	N	Y‡	N	Y‡	Y‡	N	6
[120]	Y	Y	Y	Y‡	Y	N	N	Y	N	N	Y	Y‡	Y	Y‡	Y	Y‡	N	8
[122]	Y	Y	Y	Y†	Y	Y	Y	N	N	N	N	N	N	Y	Y‡	Y‡	Y	8
[126]	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	Y‡	Y‡	Y	Y	N	7
[127]	Y	Y	Y	Y	Y	N	Y	Y	N	N	Y	N	Y	Y‡	Y	Y‡	N	10
[128]	Y	N	Y	Y	N	N	Y	Y	N	Y	N	N	N	Y‡	Y†	Y	N	7
[132]	Y†	N	Y	Y†	Y	N	Y‡	Y‡	N	N	Y	Y‡	Y‡	Y‡	N	Y‡	N	3
[133]	Y	Y	Y	Y	Y	N	Y‡	Y‡	N	N	Y	Y	Y	Y‡	Y	Y‡	N	9
[137]	Y	N	Y	Y	N	Y	N	Y‡	N	N	N	N	N	Y‡	Y	N	N	5
[138]	Y†	Y	Y	Y	Y	N	Y	Y‡	N	Y	Y	Y‡	Y	Y‡	Y	Y‡	N	9
[143]	Y	Y	Y	Y†	Y	N	N	Y	N	N	N	N	N	N	N	Y	N	6
[144]	Y	Y	N	N	N	N	N	Y	N	N	Y	N	Y	N	N	N	N	5
[146]	Y	Y	Y	Y	Y	N	Y‡	Y	N	N	Y	Y‡	Y	Y	Y	Y‡	N	10
[147]	Y	Y	Y	Y	Y	N	Y	Y	N	N	Y	Y‡	N	Y‡	Y	Y‡	N	9
[148]	Y	Y	Y	Y†	Y	N	N	Y‡	N	N	N	N	N	Y‡	Y	N	N	5
[149]	Y	Y	Y	Y	Y	N	Y	Y‡	N	N	Y	Y	Y	Y‡	Y	Y‡	N	10
[151]	Y†	Y	Y	Y‡	Y	N	N	N	N	N	N	N	N	N	N	N	N	3
[152]	Y	N	Y	Y	Y	N	Y	Y	N	N	Y	Y	Y‡	Y‡	Y	Y‡	N	9
[156]	Y†	Y	Y	Y	Y	N	N	N	N	N	Y	Y	Y	N	Y	N	N	8
[159]	Y	Y	Y	N	N	N	Y	N	N	N	Y	Y‡	Y‡	Y‡	Y‡	Y	N	6
[160]	Y	Y	Y	Y†‡	N	N	Y	N	N	N	Y	Y	Y‡	Y‡	Y‡	Y	N	7
[164]	Y†	Y	Y	Y†	Y	N	Y	N	N	N	Y	Y	Y‡	N	Y	Y	N	8
[165]	Y	Y	Y	Y	Y	N	Y	N	N	N	Y	Y‡	Y	N	N	Y‡	N	8
[166]	Y†	N	N	Y	Y	N	Y	N	Y	Y‡	Y‡	Y‡	Y‡	Y‡	Y	Y‡	N	5
[167]	Y	Y	Y	Y	Y	N	N	N	N	N	Y	Y	Y	Y‡	Y	Y‡	Y	10
[168]	Y	Y	Y	Y	Y	N	N	N	N	N	Y	Y‡	Y‡	Y‡	Y	Y‡	N	7
[169]	Y†	Y	Y	Y	Y	N	Y	Y	Y	Y†	Y	Y‡	Y	Y‡	Y	Y‡	N	12
[170]	Y	Y	Y	Y	Y	Y	Y	N	N	Y‡	Y	Y	Y	Y	Y	Y‡	N	12

Figure 7: Summary of scores that papers received in each criteria: the column titles corresponds to the item numbers used in Section III-B. In column 16, † refers to the miss-use of classification metrics together with prediction metrics by the authors, and the ‡ mark highlights when only classification metrics are employed.

calculation [168]. But when having more samples, RMSE can reconstruct error distribution, with a standard deviation lower than 5%. Inconsistency in comparing RMSEs from different studies is not due to error-scale variance alone [167].

Choosing one single metric removes a lot of information, so an error distribution should always be provided. MAE is suitable for uniformly distributed errors, while RMSE is better when errors follow a normal distribution, which is the most common case. For other kinds of distributions, more statistics, such as mean, variance, skewness, and flatness, should be provided [168]. So to better depict the model behaviour, the best recommendation might be to provide the full probability distribution of the error, or at least several standard metrics which facilitate comparisons.

When reporting results, including a statistical significance test with the p-value obtained (rather than just simply passing or not the famous 0.05 threshold) and/or confidence intervals to reflect the uncertainty in the forecast is strongly recommended. But also, in order to test if a proposal makes sense or not, it is essential to use simple reference models as baselines, such as naïve or persistence forecasting models. It is very common to see how the interest that has been put in developing the proposed model is inversely proportional to the effort invested in the benchmarking models. This may lead to overoptimistic interpretations of the results, as well as an unrealistic idea of the real capabilities of the developed model.

MonteCarlo stochastic simulations seem to be a suitable practice for modeling infectious outbreaks that change across geographical areas and through time [37]. Also, hyperparameter search and sensitivity tests are strongly recommended.

According to the American Statistical Association (ASA) [169], a study is reproducible if one can take the original data and the computer code used to analyze the data and reproduce all of the numerical findings from the study. On the other hand, replicability is the possibility of repeating an entire experiment, independently of the original investigator and without the use of original data (and generally using the same methods).

Although it might be argued that full replicability is theoretically not achievable, a clear description of the methods, models, materials, procedures, metrics, and other variables involved in the study would facilitate it. A clear description of the dataset, data pre-processing, and missing data handling is essential. A description of the statistical inference decisions made and whether the study is exploratory or confirmatory, as well as discussion of the expected constraints for generality, uncertainty of the measurements, results, and inferences are definitely helpful.

Furthermore, while the easiest way to replicate an experiment in DL is to count with the full source code and the original dataset employed, a potential opacity might occur when publicly available datasets or code are being updated. Therefore, it is also advisable to keep track of specific cached versions of datasets and code, so those can be correctly referenced. Many public repository sites provide tools to make this task much easier. These practices are also enabling scientific reproducibility, speeding up future discoveries in any discipline.

VI. Conclusions

In this systematic review, current deep learning literature for COVID-19 forecasting has been considered. We focused on evaluating a set of papers, underlining the quality flaws of the methods employed and the reproducibility and replicability issues.

After establishing a set of minimum quality indicators, it has been observed that no papers in the reviewed literature currently have documented satisfactorily the methodologies employed for the entire process, failing to follow good practices for developing a reproducible deep learning model. A common pitfall is the lack of a robust cross-validation methodology. There is a lot of room for improvement in model comparison against naïve or persistence baselines, as well as the extended use of any kind of statistical inference, to minimally discard any possibility of changes in the results. The different kinds of error metrics presented in the analyzed papers, the variety of forecast periods, and the different kinds of variables to predict, render comparisons difficult.

We agree with [19] that it is vital to develop a standardized reporting protocol and checklists to reduce the poorly conducted COVID-19 studies in favor of more properly conducted studies, and to improve replicability. Finally, some specific recommendations to the researchers for better practices regarding all the analyzed criteria have been provided.

References

[1] M. Castro *et al.*, "The turning point and end of an expanding epidemic cannot be precisely forecast," en, *Proceedings of the National Academy of Sciences*, vol. 117, no. 42, pp. 26 190–26 196, Oct. 20, 2020, publisher: National Academy of Sciences section: Biological Sciences PMID: 33004629.

[2] A. Adiga *et al.*, "Mathematical models for covid-19 pandemic: A comparative analysis," en, *Journal of the Indian Institute of Science*, vol. 100, no. 4, pp. 793–807, Oct. 1, 2020, Company: Springer Distributor: Springer Institution: Springer Label: Springer number: 4 publisher: Springer India.

[3] D. Lazer *et al.*, "The parable of google flu: Traps in big data analysis," en, *Science*, vol. 343, no. 6176, pp. 1203–1205, Mar. 14, 2014.

[4] J. W. Schooler, "Metascience could rescue the "replication crisis"," *Nature*, vol. 515, no. 7525, pp. 9–9, Nov. 2014.

[5] W. Naudé, "Artificial intelligence against covid-19: An early review," en, Social Science Research Network, Rochester, NY, Tech. Rep., Apr. 6, 2020, [Online; accessed 2021-01-24].

[6] D. C. Nguyen *et al.*, "Blockchain and ai-based solutions to combat coronavirus (covid-19)-like epidemics: A survey," en, *Preprint*, Apr. 19, 2020, publisher: Preprints.

[7] Q. Pham *et al.*, "Artificial intelligence (ai) and big data for coronavirus (covid-19) pandemic: A survey on the state-of-the-arts," *IEEE Access*, vol. 8, pp. 130 820–130 839, Apr. 21, 2020, event: IEEE Access.

[8] N. L. Bragazzi *et al.*, "How big data and artificial intelligence can help better manage the covid-19 pandemic," eng, *International Journal of Environmental Research and Public Health*, vol. 17, no. 9, May 2, 2020, PMID: 32370204 PMCID: PMC7246824.

[9] P. N. Mahalle *et al.*, "Data analytics: Covid-19 prediction using multimodal data," en, *Preprint*, May 14, 2020, publisher: Preprints.

[10] T. Alamo *et al.*, "Data-driven methods to monitor, model, forecast and control covid-19 pandemic: Leveraging data science, epidemiology and control theory," *arXiv:2006.01731 [physics, q-bio]*, Jun. 10, 2020, arXiv: 2006.01731.

[11] G. R. Shinde *et al.*, "Forecasting models for coronavirus disease (covid-19): A survey of the state-of-the-art," en, *SN Computer Science*, vol. 1, no. 4, p. 197, Jun. 11, 2020.

[12] R. Vaishya *et al.*, "Artificial intelligence (ai) applications for covid-19 pandemic," en, *Diabetes & Metabolic Syndrome: Clinical ResearchReviews*, vol. 14, no. 4, pp. 337–339, Jul. 1, 2020.

[13] J. Chen *et al.*, "A survey on applications of artificial intelligence in fighting against covid-19," *arXiv:2007.02202 [cs, q-bio]*, Jul. 4, 2020, arXiv: 2007.02202.

[14] A. Sufian *et al.*, "A survey on deep transfer learning to edge computing for mitigating the covid-19 pandemic," *Journal of Systems Architecture*, vol. 108, p. 101 830, Sep. 2020, PMID: null PMCID: PMC7326453.

[15] W. Naudé, "Artificial intelligence vs covid-19: Limitations, constraints and pitfalls," en, *AISOCIETY*, vol. 35, no. 3, pp. 761–765, Sep. 1, 2020.

[16] S. Lalmuanawma *et al.*, "Applications of machine learning and artificial intelligence for covid-19 (sars-cov-2) pandemic: A review," en, *Chaos, Solitons & Fractals*, vol. 139, p. 110 059, Oct. 1, 2020.

[17] P. Sarosh *et al.*, "Artificial intelligence for covid-19 detection – a state-of-the-art review," *arXiv:2012.06310 [cs]*, Nov. 25, 2020, arXiv: 2012.06310.

[18] J. Rasheed *et al.*, "A survey on artificial intelligence approaches in supporting frontline workers and decision makers for the covid-19 pandemic," en, *Chaos, Solitons & Fractals*, vol. 141, p. 110 337, Dec. 1, 2020.

[19] A. Abd-Alrazaq *et al.*, "Artificial intelligence in the fight against covid-19: Scoping review," eng, *Journal of Medical Internet Research*, vol. 22, no. 12, e20756, Dec. 15, 2020, PMID: 33284779 PMCID: PMC7744141.

[20] M.-H. Tayarani N., "Applications of artificial intelligence in battling against covid-19: A literature review," en, *Chaos, Solitons & Fractals*, vol. 142, p. 110 338, Jan. 1, 2021.

[21] J. Bullock *et al.*, "Mapping the landscape of artificial intelligence applications against covid-19," *arXiv:2003.11336 [cs]*, Jan. 11, 2021, arXiv: 2003.11336.

[22] J. Devaraj *et al.*, "Forecasting of covid-19 cases using deep learning models: Is it reliable and practically significant?" en, *Results in Physics*, p. 103 817, Jan. 14, 2021.

[23] A. Alimadadi *et al.*, "Artificial intelligence and machine learning to fight covid-19," *Physiological Genomics*, vol. 52, no. 4, pp. 200–202, Mar. 27, 2020, publisher: American Physiological Society.

[24] F. Shi *et al.*, "Review of artificial intelligence techniques in imaging data acquisition, segmentation and diagnosis for covid-19," English, *IEEE Reviews in Biomedical Engineering*, 2020, publisher: Institute of Electrical and Electronics Engineers Inc.

[25] Y. Mohamadou *et al.*, "A review of mathematical modeling, artificial intelligence and datasets used in the study, prediction and management of covid-19," en, *Appl Intell*, 2020, [Online; accessed 2021-01-30].

[26] A. Kumar *et al.*, "A review of modern technologies for tackling covid-19 pandemic," eng, *DiabetesMetabolic Syndrome*, vol. 14, no. 4, pp. 569–573, Aug. 2020, PMID: 32413821 PMCID: PMC7204706.

[27] M. Tsikala Vafea *et al.*, "Emerging technologies for use in the study, diagnosis, and treatment of patients with covid-19," en, *Cellular and Molecular Bioengineering*, vol. 13, no. 4, pp. 249–257, Aug. 1, 2020.

[28] A. S. Ahuja *et al.*, "Artificial intelligence and covid-19: A multidisciplinary approach," en, *Integrative Medicine Research*, Integrative Medicine for COVID-19: Researches and Evidence, vol. 9, no. 3, p. 100 434, Sep. 1, 2020.

[29] A. Shoeibi *et al.*, "Automated detection and forecasting of covid-19 using deep learning techniques: A review," *arXiv:2007.10785 [cs, eess]*, Jul. 27, 2020, arXiv: 2007.10785.

[30] R. Madurai Elavarasan and R. Pugazhendhi, "Restructured society and environment: A review on potential technological strategies to control the covid-19 pandemic," en, *Science of The Total Environment*, vol. 725, p. 138 858, Jul. 10, 2020.

[31] A. Waleed Salehi *et al.*, "Review on machine and deep learning models for the detection and prediction of coronavirus," eng, *Materials Today. Proceedings*, vol. 33, pp. 3896–3901, 2020, PMID: 32837918 PMCID: PMC7309744.

[32] S. Latif *et al.*, "Leveraging data science to combat covid-19: A comprehensive review," *IEEE Transactions on Artificial Intelligence*, vol. 1, no. 1, pp. 85–103, Aug. 2020, event: IEEE Transactions on Artificial Intelligence.

[33] L. Wynants *et al.*, "Prediction models for diagnosis and prognosis of covid-19: Systematic review and critical appraisal," *BMJ*, vol. 369, 2020, [Online; accessed 2021-01-30].

[34] Z. Hu *et al.*, “Artificial intelligence forecasting of covid-19 in china,” en, *Preprint*, Feb. 17, 2020, [Online; accessed 2020-12-15].

[35] Z. Hu *et al.*, “Forecasting and evaluating intervention of covid-19 in the world,” *arXiv:2003.09800 [q-bio]*, Mar. 21, 2020, arXiv: 2003.09800.

[36] Z. Yang *et al.*, “Modified seir and ai prediction of the epidemics trend of covid-19 in china under public health interventions,” eng, *Journal of Thoracic Disease*, vol. 12, no. 3, pp. 165–174, Mar. 2020, PMID: 32274081 PMCID: PMC7139011.

[37] S. J. Fong *et al.*, “Composite monte carlo decision making under high uncertainty of novel coronavirus epidemic using hybridized deep learning and fuzzy rule induction,” en, *Applied Soft Computing*, vol. 93, p. 106 282, Aug. 1, 2020.

[38] S. J. Fong *et al.*, “Finding an accurate early forecasting model from small dataset: A case of 2019-ncov novel coronavirus outbreak,” en, *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. Special Issue on Soft Computing, 2020, [Online; accessed 2020-12-15].

[39] R. M. Rizk-Allah and A. E. Hassanien, “Covid-19 forecasting based on an improved interior search algorithm and multi-layer feed forward neural network,” *arXiv:2004.05960 [cs, eess]*, Apr. 6, 2020, arXiv: 2004.05960.

[40] O. Torrealba-Rodriguez *et al.*, “Modeling and prediction of covid-19 in mexico applying mathematical and computational models,” en, *Chaos, Solitons & Fractals*, vol. 138, p. 109 946, Sep. 1, 2020.

[41] V. K. R. Chimmula and L. Zhang, “Time series forecasting of covid-19 transmission in canada using lstm networks,” en, *Chaos, Solitons & Fractals*, vol. 135, p. 109 864, Jun. 1, 2020.

[42] A. Tomar and N. Gupta, “Prediction for the spread of covid-19 in india and effectiveness of preventive measures,” en, *Science of The Total Environment*, vol. 728, p. 138 762, Aug. 1, 2020.

[43] N. Zheng *et al.*, “Predicting covid-19 in china using hybrid ai model,” *IEEE Transactions on Cybernetics*, 2020.

[44] C.-J. Huang *et al.*, “Multiple-input deep convolutional neural network model for covid-19 forecasting in china,” en, *medRxiv*, p. 2020.03.23.20041608, Mar. 27, 2020, publisher: Cold Spring Harbor Laboratory Press.

[45] M. A. A. Al-qaness *et al.*, “Optimization method for forecasting confirmed cases of covid-19 in china,” en, *Journal of Clinical Medicine*, vol. 9, no. 3, p. 674, Mar. 2020, number: 3 publisher: Multidisciplinary Digital Publishing Institute.

[46] S. Ayyoubzadeh *et al.*, “Predicting covid-19 incidence through analysis of google trends data in iran: Data mining and deep learning pilot study,” *JMIR Public Health and Surveillance*, vol. 6, Mar. 21, 2020.

[47] N. S. Punn *et al.*, “Covid-19 epidemic analysis using machine learning and deep learning algorithms,” en, *medRxiv*, p. 2020.04.08.20057679, Jun. 1, 2020, publisher: Cold Spring Harbor Laboratory Press.

[48] S. k. Paul *et al.*, “A multivariate spatiotemporal spread model of covid-19 using ensemble of convlstm networks,” *medRxiv*, 2020.

[49] A. Zeroual *et al.*, “Deep learning methods for forecasting covid-19 time-series data: A comparative study,” en, *Chaos, Solitons & Fractals*, vol. 140, p. 110 121, Nov. 1, 2020.

[50] R. Dandekar and G. Barbastathis, “Neural network aided quarantine control model estimation of global covid-19 spread,” *arXiv:2004.02752 [physics, q-bio]*, Apr. 2, 2020, arXiv: 2004.02752.

[51] M. R. Ibrahim *et al.*, “Variational-lstm autoencoder to forecast the spread of coronavirus across the globe,” en, *medRxiv*, p. 2020.04.20.20070938, Apr. 24, 2020, publisher: Cold Spring Harbor Laboratory Press.

[52] R. Kafieh *et al.*, “Covid-19 in iran: A deeper look into the future,” en, *medRxiv*, p. 2020.04.24.20078477, Apr. 27, 2020, publisher: Cold Spring Harbor Laboratory Press.

[53] L. R. Kolozsvari *et al.*, “Predicting the epidemic curve of the coronavirus (sars-cov-2) disease (covid-19) using artificial intelligence,” en, *medRxiv*, p. 2020.04.17.20069666, Jan. 27, 2021, publisher: Cold Spring Harbor Laboratory Press.

[54] S. Dutta and S. K. Bandyopadhyay, “Machine learning approach for confirmation of covid-19 cases: Positive, negative, death and release,” en, *medRxiv*, p. 2020.03.25.20043505, Mar. 30, 2020, publisher: Cold Spring Harbor Laboratory Press.

[55] Z. Car *et al.*, “Modeling the spread of covid-19 infection using a multilayer perceptron,” Engels, *Computational and Mathematical Methods in Medicine*, vol. 2020, Jan. 1, 2020, [Online; accessed 2021-02-01].

[56] P. Hartono, “Similarity maps and pairwise predictions for transmission dynamics of covid-19 with neural networks,” en, *Informatics in Medicine Unlocked*, vol. 20, p. 100 386, Jan. 1, 2020.

[57] S. Uhlig *et al.*, “Modeling projections for covid-19 pandemic by combining epidemiological, statistical, and neural network approaches,” en, *medRxiv*, p. 2020.04.17.20059535, Apr. 22, 2020, publisher: Cold Spring Harbor Laboratory Press.

[58] H. Dutta, “Neural network model for prediction of covid-19 confirmed cases and fatalities,” *Preprint*, May 1, 2020.

[59] B. Yan *et al.*, “An improved method for the fitting and prediction of the number of covid-19 confirmed cases based on lstm,” en, *Computers, MaterialsContinua*, vol. 64, no. 3, pp. 1473–1490, Jun. 30, 2020.

[60] M. A. A. Al-qaness *et al.*, “Marine predators algorithm for forecasting confirmed cases of covid-19 in italy, usa, iran and korea,” eng, *International Journal of Environmental Research and Public Health*, vol. 17, no. 10, May 18, 2020, PMID: 32443476 PMCID: PMC7277148.

[61] M. Karimuzzaman *et al.*, “Forecasting the covid-19 pandemic with climate variables for top five burdening and three south asian countries,” en, *medRxiv*, p. 2020.05.12.20099044, May 19, 2020, publisher: Cold Spring Harbor Laboratory Press.

[62] S. Cabras, “A bayesian - deep learning model for estimating covid-19 evolution in spain,” *arXiv:2005.10335 [stat]*, May 20, 2020, arXiv: 2005.10335.

[63] A. M. Javid *et al.*, “Predictive analysis of covid-19 time-series data from johns hopkins university,” *arXiv:2005.05060 [cs, eess]*, May 22, 2020, arXiv: 2005.05060.

[64] M. Azarafza *et al.*, “Covid-19 infection forecasting based on deep learning in iran,” en, *medRxiv*, p. 2020.05.16.20104182, May 24, 2020, publisher: Cold Spring Harbor Laboratory Press.

[65] S. M. Zandavi *et al.*, “Forecasting the spread of covid-19 under control scenarios using lstm and dynamic behavioral models,” *arXiv:2005.12270 [physics]*, May 24, 2020, arXiv: 2005.12270.

[66] C. Direkoglu and M. Sah, “Worldwide and regional forecasting of coronavirus (covid-19) spread using a deep learning model,” en, *medRxiv*, p. 2020.05.23.20111039, May 26, 2020, publisher: Cold Spring Harbor Laboratory Press.

[67] L.-P. Chen, “Analysis and prediction of covid-19 data in taiwan,” en, Social Science Research Network, Rochester, NY, Tech. Rep., May 27, 2020, DOI: 10.2139/ssrn.3611761.

[68] A. Chatterjee *et al.*, “Statistical explorations and univariate timeseries analysis on covid-19 datasets to understand the trend of disease spreading and death,” *Sensors (Basel, Switzerland)*, vol. 20, no. 11, May 29, 2020, PMID: 32486055 PMCID: PMC7308840.

[69] G. Pinter *et al.*, “Covid-19 pandemic prediction for hungary; a hybrid machine learning approach,” *Mathematics*, vol. 8, no. 6, 2020.

[70] T. H. H. Aldhyani *et al.*, “Deep learning and holt-trend algorithms for predicting covid-19 pandemic,” en, *medRxiv*, p. 2020.06.03.20121590, Jun. 5, 2020, publisher: Cold Spring Harbor Laboratory Press.

[71] R. S. Pontoh *et al.*, “Effectiveness of the public health measures to prevent the spread of covid-19,” en, *Commun. Math. Biol. Neurosci.*, vol. 2020, no. 0, Article ID 31, Jun. 18, 2020, number: 0.

[72] P. Melin *et al.*, “Multiple ensemble neural network models with fuzzy response aggregation for predicting covid-19 time series: The case of mexico,” eng, *Healthcare (Basel, Switzerland)*, vol. 8, no. 2, Jun. 19, 2020, PMID: 32575622 PMCID: PMC7349072.

[73] S. R. Vadyala *et al.*, “Prediction of the number of covid-19 confirmed cases based on k-means-lstm,” *arXiv:2006.14752 [physics, q-bio]*, Jun. 25, 2020, arXiv: 2006.14752.

[74] Y. Tian *et al.*, “Forecasting covid-19 cases using machine learning models,” en, *medRxiv*, p. 2020.07.02.20145474, Jul. 4, 2020, publisher: Cold Spring Harbor Laboratory Press.

[75] A. Kapoor *et al.*, “Examining covid-19 forecasting using spatio-temporal graph neural networks,” *arXiv:2007.03113 [cs]*, Jul. 6, 2020, arXiv: 2007.03113.

[76] L. Moftakhar *et al.*, “Exponentially increasing trend of infected patients with covid-19 in iran: A comparison of neural network and arima forecasting models,” *Iranian Journal of Public Health*, vol. 49, Jul. 11, 2020.

[77] S. K. Tamang *et al.*, “Forecasting of covid-19 cases based on prediction using artificial neural network curve fitting technique,” *Global Journal of Environmental Science and Management*, vol. 6, no. Special Issue (Covid-19), pp. 53–64, Aug. 1, 2020.

[78] N. Hasan, “A methodological approach for predicting covid-19 epidemic using eemd-ann hybrid model,” en, *Internet of Things*, vol. 11, p. 100 228, Sep. 1, 2020.

[79] R. G. da Silva *et al.*, “Forecasting brazilian and american covid-19 cases based on artificial intelligence coupled with climatic exogenous variables,” *Chaos, Solitons, and Fractals*, vol. 139, p. 110 027, Oct. 2020, PMID: 32834591 PMCID: PMC7324930.

[80] T. B. Alakus and I. Turkoglu, “Comparison of deep learning approaches to predict covid-19 infection,” en, *Chaos, Solitons & Fractals*, vol. 140, p. 110 120, Nov. 1, 2020.

[81] M. Roberts *et al.*, “Common pitfalls and recommendations for using machine learning to detect and prognosticate for covid-19 using chest radiographs and ct scans,” en, *Nature Machine Intelligence*, vol. 3, no. 3, pp. 199–217, Mar. 2021, number: 3 publisher: Nature Publishing Group.

[82] *Strategies for the surveillance of covid-19*, en, Available at https://www.ecdc.europa.eu/en/publications-data/strategies-surveillance-covid-19 [Online; accessed 2021-02-28], Apr. 9, 2020.

[83] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” en, *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, Dec. 1, 1943, Company: Springer Distributor: Springer Institution: Springer Label: Springer number: 4 publisher: Kluwer Academic Publishers.

[84] J. Patterson and A. Gibson, *Deep Learning*, en. O’Reilly Media, Inc., Aug. 2017.

[85] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain,” eng, *Psychological Review*, vol. 65, no. 6, pp. 386–408, Nov. 1958, PMID: 13602029.

[86] ——, “Principles of neurodynamics. perceptrons and the theory of brain mechanisms,” *The American Journal of Psychology*, vol. 76, no. 4, pp. 705–707, 1963.

[87] B. Curry and P. H. Morgan, “Neural networks, linear functions and neglected non-linearity,” en, *Computational Management Science*, vol. 1, no. 1, pp. 15–29, Dec. 1, 2003.

[88] Y. Lecun, *PhD thesis: Modeles connexionnistes de l’apprentissage (connectionist learning models)*, En-

glish (US). Universite P. et M. Curie (Paris 6), Jun. 1987.

[89] D. Ballard, “Modular learning in neural networks,” *AAAI 87*, pp. 279–284, 1987.

[90] P. J. Werbos, “Beyond regression: New tools for prediction and analysis in the behavioral sciences,” PhD thesis, Harvard University, 1974.

[91] D. E. Rumelhart *et al.*, “Learning representations by back-propagating errors,” en, *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, number: 6088 publisher: Nature Publishing Group.

[92] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, pp. 1735–80, Dec. 1, 1997.

[93] K. Cho *et al.*, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv:1406.1078 [cs, stat]*, Sep. 2, 2014, arXiv: 1406.1078.

[94] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997, event: IEEE Transactions on Signal Processing.

[95] A. Vaswani *et al.*, “Attention is all you need,” *arXiv:1706.03762 [cs]*, Dec. 5, 2017, arXiv: 1706.03762 version: 5.

[96] Y. Lecun, “Generalization and network design strategies,” English (US), *Connectionism in perspective*, 1989, publisher: Elsevier.

[97] G.-B. Huang *et al.*, “Extreme learning machine: A new learning scheme of feedforward neural networks,” in *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, vol. 2, 2004, 985–990 vol.2.

[98] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” en, *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 28, 2006, publisher: American Association for the Advancement of Science section: Report PMID: 16873662.

[99] S. H. Park and K. Han, “Methodologic guide for evaluating clinical performance and effect of artificial intelligence technology for medical diagnosis and prediction,” *Radiology*, vol. 286, no. 3, pp. 800–809, Jan. 8, 2018, publisher: Radiological Society of North America.

[100] S. H. Park and H. Y. Kressel, “Connecting technological innovation in artificial intelligence to real-world medical practice through rigorous clinical validation: What peer-reviewed medical journals could do,” eng, *Journal of Korean Medical Science*, vol. 33, no. 22, e152, May 28, 2018, PMID: 29805337 PMCID: PMC5966371.

[101] G. S. Handelman *et al.*, “Peering into the black box of artificial intelligence: Evaluation metrics of machine learning methods,” *American Journal of Roentgenology*, vol. 212, no. 1, pp. 38–43, Oct. 17, 2018, publisher: American Roentgen Ray Society.

[102] D. A. Bluemke *et al.*, “Assessing radiology research on artificial intelligence: A brief guide for authors, reviewers, and readers—from the radiology editorial board,” *Radiology*, vol. 294, no. 3, pp. 487–489, Dec. 31, 2019, publisher: Radiological Society of North America.

[103] D. G. Altman *et al.*, “Equator: Reporting guidelines for health research,” English, *The Lancet*, vol. 371, no. 9619, pp. 1149–1150, Apr. 5, 2008, publisher: Elsevier PMID: 18395566.

[104] J. M. Provenzale and R. J. Stanley, “A systematic guide to reviewing a manuscript,” eng, *AJR. American journal of roentgenology*, vol. 185, no. 4, pp. 848–854, Oct. 2005, PMID: 16177399.

[105] W. Luo *et al.*, “Guidelines for developing and reporting machine learning predictive models in biomedical research: A multidisciplinary view,” EN, *Journal of Medical Internet Research*, vol. 18, no. 12, e5870, Dec. 16, 2016, Company: Journal of Medical Internet Research Distributor: Journal of Medical Internet Research Institution: Journal of Medical Internet Research Label: Journal of Medical Internet Research publisher: JMIR Publications Inc., Toronto, Canada.

[106] J. Mongan *et al.*, “Checklist for artificial intelligence in medical imaging (claim): A guide for authors and reviewers,” *Radiology: Artificial Intelligence*, vol. 2, no. 2, e200029, Mar. 1, 2020, publisher: Radiological Society of North America.

[107] R. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice (3rd ed)*, 3rd ed. Melbourne, Australia: OTexts, 2021, [Online; accessed 2021-04-14].

[108] S. K. Smith and T. Sincich, “An empirical analysis of the effect of length of forecast horizon on population forecast errors,” *Demography*, vol. 28, no. 2, pp. 261–274, 1991, publisher: Springer.

[109] Y. Wang *et al.*, “The influence of the activation function in a convolution neural network model of facial expression recognition,” en, *Applied Sciences*, vol. 10, no. 5, p. 1897, Jan. 2020, number: 5 publisher: Multidisciplinary Digital Publishing Institute.

[110] I. Goodfellow *et al.*, *Deep Learning*. The MIT Press, 2016.

[111] G. Perin and S. Picek, “On the influence of optimizers in deep learning-based side-channel analysis,” *IACR Cryptol. ePrint Arch.*, 2020.

[112] C. Bergmeir *et al.*, “A note on the validity of cross-validation for evaluating autoregressive time series prediction,” *Computational Statistics & Data Analysis*, vol. 120, pp. 70–83, 2018.

[113] E. Steyerberg, *Clinical Prediction Models: A Practical Approach to Development, Validation, and Updating*, en, 2nd ed., ser. Statistics for Biology and Health. Springer International Publishing, 2019, DOI: 10.1007/978-3-030-16399-0.

[114] V. Amrhein *et al.*, “Scientists rise up against statistical significance,” en, *Nature*, vol. 567, no. 7748, pp. 305–307, Mar. 2019, number: 7748 publisher: Nature Publishing Group.

[115] D. W. Hosmer *et al.*, "A comparison of goodness-of-fit tests for the logistic regression model," en, *Statistics in Medicine*, vol. 16, no. 9, pp. 965–980, 1997.

[116] R. Nuzzo, "Scientific method: Statistical errors," en, *Nature News*, vol. 506, no. 7487, p. 150, Feb. 13, 2014, section: News Feature.

[117] A. Davydenko and R. Fildes, *Forecast error measures: Critical review and practical recommendations*, Jan. 2016.

[118] *Retracted coronavirus (covid-19) papers*, en-US, Available at https://retractionwatch.com/retracted-coronavirus-covid-19-papers [Online; accessed 2021-04-27], Apr. 29, 2020.

[119] N. B. Yahia *et al.*, "Deep ensemble learning method to forecast covid-19 outbreak," In Review, Tech. Rep., May 21, 2020, DOI: 10.21203/rs.3.rs-27216/v1.

[120] N. Yudistira, "Covid-19 growth prediction using multivariate long short term memory," en, *Preprint*, May 10, 2020, [Online; accessed 2020-12-15].

[121] A. Chatterjee and S. Roy, "An analytics overview & lstm-based predictive modeling of covid-19: A hardheaded look across india," in *Machine Intelligence and Soft Computing*, D. Bhattacharyya and N. Thirupathi Rao, Eds., Singapore: Springer Singapore, 2021, pp. 289–307.

[122] F. Shahid *et al.*, "Predictions for covid-19 with deep learning models of lstm, gru and bi-lstm," en, *Chaos, Solitons & Fractals*, vol. 140, p. 110 212, Nov. 1, 2020.

[123] L. Mohimont *et al.*, "Convolutional neural networks and temporal cnns for covid-19 forecasting in france," en, *Applied Intelligence*, 2021, [Online; accessed 2021-01-08].

[124] H. Abbasimehr and R. Paki, "Prediction of covid-19 confirmed cases combining deep learning methods and bayesian optimization," en, *Chaos, Solitons & Fractals*, vol. 142, p. 110 511, Jan. 1, 2021.

[125] V. Bharadi, "Random net implementation of mlp and lstms using averaging ensembles of deep learning models," *2020 International Conference on Decision Aid Sciences and Application (DASA)*, pp. 1197–1204, 2020.

[126] A. Prakash *et al.*, "Spread peak prediction of covid-19 using ann and regression (workshop paper)," in *2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM)*, 2020, pp. 356–365.

[127] A. Mollalo *et al.*, "Artificial neural network modeling of novel coronavirus (covid-19) incidence rates across the continental united states," en, *International Journal of Environmental Research and Public Health*, vol. 17, no. 12, p. 4204, Jan. 2020, number: 12 publisher: Multidisciplinary Digital Publishing Institute.

[128] M. Saqib, "Forecasting covid-19 outbreak progression using hybrid polynomial-bayesian ridge regression model," en, *Applied Intelligence*, Oct. 23, 2020, [Online; accessed 2021-01-22].

[129] R. Moulay Taj *et al.*, "Towards using recurrent neural networks for predicting influenza-like illness: Case study of covid-19 in morocco," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, pp. 7945–7950, Oct. 22, 2020.

[130] K. Shyam Sunder Reddy *et al.*, "Recurrent neural network based prediction of number of covid-19 cases in india," en, *Materials Today: Proceedings*, Nov. 17, 2020, [Online; accessed 2021-01-12].

[131] M. A. M. Arceda *et al.*, "Forecasting time series with multiplicative trend exponential smoothing and lstm: Covid-19 case study," in *Proceedings of the Future Technologies Conference (FTC) 2020, Volume 2*, K. Arai *et al.*, Eds., Cham: Springer International Publishing, 2021, pp. 568–582.

[132] S. Chander *et al.*, "Jaya spider monkey optimization-driven deep convolutional lstm for the prediction of covid'19," *Bio-Algorithms and Med-Systems*, vol. 16, Nov. 13, 2020.

[133] A. S. Fokas *et al.*, "Mathematical models and deep learning for predicting the number of individuals reported to be infected with sars-cov-2," *Journal of The Royal Society Interface*, vol. 17, no. 169, p. 20 200 494, 2020.

[134] S. Chakraborty *et al.*, "Reaction order and neural network approaches for the simulation of covid-19 spreading kinetic in india," en, *Infectious Disease Modelling*, vol. 5, pp. 737–747, Jan. 1, 2020.

[135] Z. Li *et al.*, "A recurrent neural network and differential equation based spatiotemporal infectious disease model with application to covid-19," *arXiv:2007.10929 [cs, q-bio, stat]*, Sep. 17, 2020, arXiv: 2007.10929.

[136] M. Wieczorek *et al.*, "Neural network powered covid-19 spread forecasting model," en, *Chaos, Solitons & Fractals*, vol. 140, p. 110 203, Nov. 1, 2020.

[137] I. Pereira *et al.*, "Forecasting covid-19 dynamics in brazil: A data driven approach," *International Journal of Environmental Research and Public Health*, vol. 17, p. 5115, Jul. 15, 2020.

[138] İ. Kırbaş *et al.*, "Comparative analysis and forecasting of covid-19 cases in various european countries with arima, narnn and lstm approaches," en, *Chaos, Solitons & Fractals*, vol. 138, p. 110 015, Sep. 1, 2020.

[139] M. Wieczorek *et al.*, "Real-time neural network based predictor for cov19 virus spread," en, *PLOS ONE*, vol. 15, no. 12, e0243189, 2020, publisher: Public Library of Science.

[140] A. Rodriguez *et al.*, "Deepcovid: An operational deep learning-driven framework for explainable real-time covid-19 forecasting," en, *medRxiv*, p. 2020.09.28.20203109, Sep. 29, 2020, publisher: Cold Spring Harbor Laboratory Press.

[141] A. Ramchandani *et al.*, "Deepcovidnet: An interpretable deep learning model for predictive surveillance of covid-19 using heterogeneous features and their interactions," *IEEE Access*, vol. 8, pp. 159 915–159 930, 2020, event: IEEE Access.

[142] R. Sujath *et al.*, "A machine learning forecasting model for covid-19 pandemic in india," en, *Stochastic Environmental Research and Risk Assessment*,

vol. 34, no. 7, pp. 959–972, Jul. 1, 2020, Company: Springer Distributor: Springer Institution: Springer Label: Springer number: 7 publisher: Springer Berlin Heidelberg.

[143] S. D. Khan *et al.*, "Toward smart lockdown: A novel approach for covid-19 hotspots prediction using a deep hybrid neural network," en, *Computers*, vol. 9, no. 4, p. 99, Dec. 11, 2020.

[144] C. Distante *et al.*, "Forecasting covid-19 outbreak progression in italian regions: A model based on neural network training from chinese data," en, *medRxiv*, p. 2020.04.09.20059055, Apr. 14, 2020, publisher: Cold Spring Harbor Laboratory Press.

[145] A. H. Elsheikh *et al.*, "Deep learning-based forecasting model for covid-19 outbreak in saudi arabia," en, *Process Safety and Environmental Protection*, vol. 149, pp. 223–233, May 1, 2021.

[146] S. Dhamodharavadhani *et al.*, "Covid-19 mortality rate prediction for india using statistical neural network models," English, *Frontiers in Public Health*, vol. 8, 2020, publisher: Frontiers.

[147] S. Prasanth *et al.*, "Forecasting spread of covid-19 using google trends: A hybrid gwo-deep learning approach," en, *Chaos, Solitons & Fractals*, vol. 142, p. 110 336, Jan. 1, 2021.

[148] H. T. Rauf *et al.*, "Time series forecasting of covid-19 transmission in asia pacific countries using deep neural networks," en, *Personal and Ubiquitous Computing*, Jan. 10, 2021, [Online; accessed 2021-01-22].

[149] P. Arora *et al.*, "Prediction and analysis of covid-19 positive cases using deep learning models: A descriptive case study of india," en, *Chaos, Solitons & Fractals*, vol. 139, p. 110 017, Oct. 1, 2020.

[150] O. Istaiteh *et al.*, "Machine learning approaches for covid-19 forecasting," in *2020 International Conference on Intelligent Data Science Technologies and Applications (IDSTA)*, 2020, pp. 50–57.

[151] Y. Karadayi *et al.*, "Unsupervised anomaly detection in multivariate spatio-temporal data using deep learning: Early detection of covid-19 outbreak in italy," *IEEE Access*, vol. 8, pp. 164 155–164 177, 2020, event: IEEE Access.

[152] A. I. Saba and A. H. Elsheikh, "Forecasting the prevalence of covid-19 outbreak in egypt using nonlinear autoregressive artificial neural networks," en, *Process Safety and Environmental Protection*, vol. 141, pp. 1–8, Sep. 1, 2020.

[153] S. Ballı, "Data analysis of covid-19 pandemic and short-term cumulative case forecasting using machine learning time series methods," en, *Chaos, Solitons & Fractals*, vol. 142, p. 110 512, Jan. 1, 2021.

[154] K. T. Ly, "A covid-19 forecasting system using adaptive neuro-fuzzy inference," en, *Finance Research Letters*, p. 101 844, Nov. 12, 2020.

[155] V. Papastefanopoulos *et al.*, "Covid-19: A comparison of time series methods to forecast percentage of active cases per population," en, *Applied Sciences*, vol. 10, no. 11, p. 3880, Jan. 2020, number: 11 publisher: Multidisciplinary Digital Publishing Institute.

[156] M. Hawas, "Generated time-series prediction data of covid-19's daily infections in brazil by using recurrent neural networks," en, *Data in Brief*, vol. 32, p. 106 175, Oct. 1, 2020.

[157] M. A. Achterberg *et al.*, "Comparing the accuracy of several network-based covid-19 prediction algorithms," en, *International Journal of Forecasting*, Oct. 9, 2020, [Online; accessed 2021-01-20].

[158] P. Wang *et al.*, "Time series prediction for the epidemic trends of covid-19 using the improved lstm deep learning method: Case studies in russia, peru and iran," en, *Chaos, Solitons & Fractals*, vol. 140, p. 110 214, Nov. 1, 2020.

[159] S. Thakur *et al.*, "Prediction for the second wave of covid-19 in india," in *Big Data Analytics*, L. Bellatreche *et al.*, Eds., Cham: Springer International Publishing, 2020, pp. 134–150.

[160] S. Shastri *et al.*, "Time series forecasting of covid-19 using deep learning models: India-usa comparative case study," en, *Chaos, Solitons & Fractals*, vol. 140, p. 110 227, Nov. 1, 2020.

[161] S. Saif *et al.*, "A hybrid model based on mba-anfis for covid-19 confirmed cases prediction and forecast," en, *Journal of The Institution of Engineers (India): Series B*, Jan. 19, 2021, [Online; accessed 2021-01-22].

[162] Z. Zhao *et al.*, "How well can we forecast the covid-19 pandemic with curve fitting and recurrent neural networks?" *Preprint*, May 18, 2020, DOI: 10.1101/2020.05.14.20102541.

[163] N. M. Ghazaly *et al.*, "Novel coronavirus forecasting model using nonlinear autoregressive artificial neural network," en, *International Journal of Advanced Science and Technology*, vol. 29, no. 5s, pp. 1831–1849, Apr. 9, 2020, number: 5s.

[164] S. Bahri *et al.*, "Deep learning for covid-19 prediction," in *2020 4th International Conference on Advanced Systems and Emergent Technologies*, 2020, pp. 406–411.

[165] Y. Gautam, "Transfer learning for covid-19 cases and deaths forecast using lstm network," en, *ISA Transactions*, Jan. 4, 2021, [Online; accessed 2021-01-20].

[166] J. A. L. Marques *et al.*, "Artificial intelligence prediction for the covid-19 data based on lstm neural networks and h2o automl," en, in *Predictive Models for Decision Support in the COVID-19 Crisis*, ser. SpringerBriefs in Applied Sciences and Technology, J. A. L. Marques *et al.*, Eds., Cham: Springer International Publishing, 2021, pp. 69–87.

[167] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance," *Climate Research*, vol. 30, no. 1, pp. 79–82, 2005, publisher: Inter-Research Science Center.

[168] T. Chai and R. Draxler, "Root mean square error (rmse) or mean absolute error (mae)?" *Geosci. Model Dev.*, vol. 7, Jan. 31, 2014.

[169] K. Broman *et al.*, *Recommendations to funding agencies for supporting reproducible research*, en.

Appendix B

Publication list

Chapter 3 Rodrigo de Medrano and José L. Aznarte. “A spatio-temporal attention-based spot-forecasting framework for urban traffic prediction”. In:Applied Soft Computing 96 (Nov. 2020), p. 106615.ISSN: 1568-4946.DOI:10.1016/j.asoc.2020.106615.

Chapter 4 Rodrigo de Medrano and José L Aznarte. “On the inclusion of spatial information for spatio-temporal neural networks”. In:Neural Computing and Applications(2021), pp. 1–18.DOI:10.1007/s00521-021-06111-6.

Chapter 5 Rodrigo de Medrano, Víctor de Buen Remiro, and José L Aznarte.“SOCAIRE: Forecasting and Monitoring Urban Air Quality in Madrid”.In: Environmental Modelling & Software(Sept.2021).DOI:10.1016/j.envsoft.2021.105084.

Chapter 6 Rodrigo de Medrano and José L Aznarte. “A New Spatio-Temporal Neural Network Approach for Traffic Accident Forecasting”. In:Applied Artificial Intelligence(2021), pp. 1–20. DOI:10.1080/08839514.2021.1935588.

Appendix A Luis Gutiérrez, Rodrigo de Medrano, and José L Aznarte. “COVID-19 forecasting with deep learning: a distressing survey”. IEEE Transactions on Neural Networks and Learning Systems. *Under Review*

Bibliography

[1] Hossein Abbasimehr and Reza Paki. “Prediction of COVID-19 confirmed cases combining deep learning methods and Bayesian optimization”. en. In: *Chaos, Solitons & Fractals* 142 (Jan. 1, 2021), p. 110511. ISSN: 0960-0779. DOI: 10.1016/j.chaos.2020.110511.

[2] Alaa Abd-Alrazaq et al. “Artificial Intelligence in the Fight Against COVID-19: Scoping Review”. eng. In: *Journal of Medical Internet Research* 22.12 (Dec. 15, 2020), e20756. ISSN: 1438-8871. DOI: 10.2196/20756.

[3] Mohamed A. Abdel-Aty and A. Essam Radwan. “Modeling traffic accident occurrence and involvement”. In: *Accident Analysis & Prevention* 32.5 (2000), pp. 633–642. ISSN: 0001-4575. DOI: 10.1016/S0001-4575(99)00094-9.

[4] Massimo A. Achterberg et al. “Comparing the accuracy of several network-based COVID-19 prediction algorithms”. en. In: *International Journal of Forecasting* (Oct. 9, 2020). ISSN: 0169-2070. DOI: 10.1016/j.ijforecast.2020.10.001.

[5] Aniruddha Adiga et al. “Mathematical Models for COVID-19 Pandemic: A Comparative Analysis”. en. In: *Journal of the Indian Institute of Science* 100.4 (Oct. 1, 2020), pp. 793–807. ISSN: 0019-4964. DOI: 10.1007/s41745-020-00200-6.

[6] AEMet. *Agencia Estatal de Meteorología - AEMET. Gobierno de España*. es. URL: http://www.aemet.es/es/portada.

[7] Abhimanyu S. Ahuja, Vineet Pasam Reddy, and Oge Marques. “Artificial intelligence and COVID-19: A multidisciplinary approach”. en. In: *Integrative Medicine Research*. Integrative Medicine for COVID-19: Researches and Evidence 9.3 (Sept. 1, 2020), p. 100434. ISSN: 2213-4220. DOI: 10.1016/j.imr.2020.100434.

[8] Yi Ai et al. “A deep learning approach on short-term spatiotemporal distribution forecasting of dockless bike-sharing system”. en. In: *Neural Computing and Applications* 31.5 (May 2019), pp. 1665–1677. ISSN: 1433-3058. DOI: 10.1007/s00521-018-3470-9.

[9] Mohammed A. A. Al-qaness et al. “Marine Predators Algorithm for Forecasting Confirmed Cases of COVID-19 in Italy, USA, Iran and Korea”. eng. In: *International Journal of Environmental Research and Public Health* 17.10 (May 18, 2020). ISSN: 1660-4601. DOI: 10.3390/ijerph17103520.

[10] Mohammed A. A. Al-qaness et al. "Optimization Method for Forecasting Confirmed Cases of COVID-19 in China". en. In: *Journal of Clinical Medicine* 9.3 (Mar. 2020), p. 674. DOI: `10.3390/jcm9030674`.

[11] Alexandre Alahi et al. "Social LSTM: Human Trajectory Prediction in Crowded Spaces". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. ISSN: 1063-6919. June 2016, pp. 961–971. DOI: `10.1109/CVPR.2016.110`.

[12] Walaa Alajali, Wanlei Zhou, and Sheng Wen. "Traffic Flow Prediction for Road Intersection Safety". In: *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation*. Guangzhou, China: IEEE, Oct. 2018, pp. 812–820. ISBN: 978-1-5386-9380-3. DOI: `10.1109/SmartWorld.2018.00151`.

[13] Talha Burak Alakus and Ibrahim Turkoglu. "Comparison of deep learning approaches to predict COVID-19 infection". en. In: *Chaos, Solitons & Fractals* 140 (Nov. 1, 2020), p. 110120. ISSN: 0960-0779. DOI: `10.1016/j.chaos.2020.110120`.

[14] Teodoro Alamo, D. G. Reina, and Pablo Millán. "Data-Driven Methods to Monitor, Model, Forecast and Control Covid-19 Pandemic: Leveraging Data Science, Epidemiology and Control Theory". In: *arXiv:2006.01731 [physics, q-bio]* (June 10, 2020).

[15] Theyazn H. H. Aldhyani et al. "Deep Learning and Holt-Trend Algorithms for predicting COVID-19 pandemic". en. In: *medRxiv* (June 5, 2020), p. 2020.06.03.20121590. DOI: `10.1101/2020.06.03.20121590`.

[16] Ahmad Alimadadi et al. "Artificial intelligence and machine learning to fight COVID-19". In: *Physiological Genomics* 52.4 (Mar. 27, 2020), pp. 200–202. ISSN: 1094-8341. DOI: `10.1152/physiolgenomics.00029.2020`.

[17] Douglas G. Altman et al. "EQUATOR: reporting guidelines for health research". English. In: *The Lancet* 371.9619 (Apr. 5, 2008), pp. 1149–1150. ISSN: 0140-6736, 1474-547X. DOI: `10.1016/S0140-6736(08)60505-X`.

[18] Valentin Amrhein, Sander Greenland, and Blake McShane. "Scientists rise up against statistical significance". en. In: *Nature* 567.7748 (Mar. 2019), pp. 305–307. DOI: `10.1038/d41586-019-00857-9`.

[19] Parham Aram, Visakan Kadirkamanathan, and Sean R. Anderson. "Spatiotemporal System Identification With Continuous Spatial Maps and Sparse Estimation". In: *IEEE Transactions on Neural Networks and Learning Systems* 26.11 (Nov. 2015), pp. 2978–2983. ISSN: 2162-2388. DOI: `10.1109/TNNLS.2015.2392563`.

[20] M. A. Machaca Arceda, P. C. Laguna Laura, and V. E. Machaca Arceda. "Forecasting Time Series with Multiplicative Trend Exponential Smoothing and LSTM: COVID-19 Case Study". In: *Proceedings of the Future Technologies Conference (FTC) 2020, Volume 2*. Ed. by Kohei Arai, Supriya Kapoor, and Rahul Bhatia. Cham: Springer International Publishing, 2021, pp. 568–582. ISBN: 978-3-030-63089-8.

[21] Sina Ardabili et al. "COVID-19 Outbreak Prediction with Machine Learning". In: *Algorithms* 13 (Sept. 1, 2020). DOI: 10.3390/a13100249.

[22] Parul Arora, Himanshu Kumar, and Bijaya Ketan Panigrahi. "Prediction and analysis of COVID-19 positive cases using deep learning models: A descriptive case study of India". en. In: *Chaos, Solitons & Fractals* 139 (Oct. 1, 2020), p. 110017. ISSN: 0960-0779. DOI: 10.1016/j.chaos.2020.110017.

[23] *Artifact Review and Badging - Current*. en. URL: https://www.acm.org/publications/policies/artifact-review-and-badging-current.

[24] Reza Asadi and Amelia C. Regan. "A spatio-temporal decomposition based deep neural network for time series forecasting". en. In: *Applied Soft Computing* 87 (Feb. 2020), p. 105963. ISSN: 1568-4946. DOI: 10.1016/j.asoc.2019.105963.

[25] Seyed Ayyoubzadeh et al. "Predicting COVID-19 Incidence Through Analysis of Google Trends Data in Iran: Data Mining and Deep Learning Pilot Study". In: *JMIR Public Health and Surveillance* 6 (Mar. 21, 2020). DOI: 10.2196/18828.

[26] Mehdi Azarafza, Mohammad Azarafza, and Jafar Tanha. "COVID-19 Infection Forecasting based on Deep Learning in Iran". en. In: *medRxiv* (May 24, 2020), p. 2020.05.16.20104182. DOI: 10.1101/2020.05.16.20104182.

[27] José L. Aznarte. "Probabilistic forecasting for extreme NO2 pollution episodes". en. In: *Environmental Pollution* 229 (Oct. 2017), pp. 321–328. ISSN: 0269-7491. DOI: 10.1016/j.envpol.2017.05.079.

[28] Artur J. Badyda, James Grellier, and Piotr Dąbrowiecki. "Ambient PM2.5 Exposure and Mortality Due to Lung Cancer and Cardiopulmonary Diseases in Polish Cities". eng. In: *Advances in Experimental Medicine and Biology* 944 (2017), pp. 9–17. ISSN: 0065-2598. DOI: 10.1007/5584_2016_55.

[29] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate". In: *International Conference on Learning Representations* (May 2015).

[30] Safa Bahri, Moetez Kdayem, and Nesrine Zoghlami. "Deep Learning for COVID-19 prediction". In: *2020 4th International Conference on Advanced Systems and Emergent Technologies*. 2020, pp. 406–411. DOI: 10.1109/IC_ASET49463.2020.9318297.

[31] Lu Bai et al. "Air Pollution Forecasts: An Overview". In: *International Journal of Environmental Research and Public Health* 15.4 (Apr. 2018). ISSN: 1661-7827. DOI: 10.3390/ijerph15040780.

[32] D. Ballard. "Modular Learning in Neural Networks". In: *AAAI 87* (1987), pp. 279–284.

[33] Serkan Ballı. "Data analysis of Covid-19 pandemic and short-term cumulative case forecasting using machine learning time series methods". en. In: *Chaos, Solitons & Fractals* 142 (Jan. 1, 2021), p. 110512. ISSN: 0960-0779. DOI: 10.1016/j.chaos.2020.110512.

[34] Kasun Bandara, Christoph Bergmeir, and Hansika Hewamalage. "LSTM-MSNet: Leveraging Forecasts on Sets of Related Time Series With Multiple Seasonal Patterns". In: *IEEE Transactions on Neural Networks and Learning Systems* 32.4 (2021), pp. 1586–1599. DOI: 10.1109/TNNLS.2020.2985720.

[35] Christoph Bergmeir, Rob J. Hyndman, and Bonsoo Koo. "A note on the validity of cross-validation for evaluating autoregressive time series prediction". en. In: *Computational Statistics & Data Analysis* 120 (Apr. 2018), pp. 70–83. ISSN: 01679473. DOI: 10.1016/j.csda.2017.11.003.

[36] Christoph Bergmeir, Rob J. Hyndman, and Bonsoo Koo. "A note on the validity of cross-validation for evaluating autoregressive time series prediction". In: *Computational Statistics & Data Analysis* 120 (2018), pp. 70–83. ISSN: 0167-9473. DOI: https://doi.org/10.1016/j.csda.2017.11.003.

[37] Christoph Bergmeir, Rob J. Hyndman, and José M. Benítez. "Bagging exponential smoothing methods using STL decomposition and Box–Cox transformation". en. In: *International Journal of Forecasting* 32.2 (Apr. 2016), pp. 303–312. ISSN: 0169-2070. DOI: 10.1016/j.ijforecast.2015.07.002.

[38] V. Bharadi. "Random Net Implementation of MLP and LSTMs Using Averaging Ensembles of Deep Learning Models". In: *2020 International Conference on Decision Aid Sciences and Application (DASA)* (2020), pp. 1197–1204.

[39] Helene Blanchonnet. *Set I - Atmospheric Model high resolution 10-day forecast (HRES)*. en. Text. Apr. 2015. URL: https://www.ecmwf.int/en/forecasts/datasets/set-i.

[40] David A. Bluemke et al. "Assessing Radiology Research on Artificial Intelligence: A Brief Guide for Authors, Reviewers, and Readers—From the Radiology Editorial Board". In: *Radiology* 294.3 (Dec. 31, 2019), pp. 487–489. ISSN: 0033-8419. DOI: 10.1148/radiol.2019192515.

[41] Toon Bogaerts et al. "A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data". en. In: *Transportation Research Part C: Emerging Technologies* 112 (Mar. 2020), pp. 62–77. ISSN: 0968-090X. DOI: 10.1016/j.trc.2020.01.010.

[42] George EP Box et al. *Time series analysis: forecasting and control*. John Wiley & Sons, 1976.

[43] Nicola Luigi Bragazzi et al. “How Big Data and Artificial Intelligence Can Help Better Manage the COVID-19 Pandemic”. eng. In: *International Journal of Environmental Research and Public Health* 17.9 (May 2, 2020). ISSN: 1660-4601. DOI: `10.3390/ijerph17093176`.

[44] Leo Breiman. “Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author)”. In: *Statist. Sci.* 16.3 (Aug. 2001), pp. 199–231. DOI: `10.1214/ss/1009213726`. URL: `https://doi.org/10.1214/ss/1009213726`.

[45] Karl Broman et al. *Recommendations to Funding Agencies for Supporting Reproducible Research*. en.

[46] Jason Brownlee. *Deep Learning for Time Series Forecasting*. Machine Learning Mastery, 2019. URL: `https://machinelearningmastery.com/deep-learning-for-time-series-forecasting/`.

[47] Joseph Bullock et al. “Mapping the Landscape of Artificial Intelligence Applications against COVID-19”. In: *arXiv:2003.11336 [cs]* (Jan. 11, 2021). DOI: `10.1613/jair.1.12162`.

[48] Stefano Cabras. “A Bayesian - Deep Learning model for estimating Covid-19 evolution in Spain”. In: *arXiv:2005.10335 [stat]* (May 20, 2020).

[49] CAMS. *Copernicus air quality monitoring*. URL: `https://atmosphere.copernicus.eu/`.

[50] CAMS. *Monitoring Atmospheric Composition and Climate -III | MACC-III Project | H2020 | CORDIS | European Commission*. URL: `https://cordis.europa.eu/project/id/633080/es`.

[51] Zlatan Car et al. “Modeling the Spread of COVID-19 Infection Using a Multilayer Perceptron”. Engels. In: *Computational and Mathematical Methods in Medicine* 2020 (Jan. 1, 2020). DOI: `10.1155/2020/5714714`.

[52] Mario Castro et al. “The turning point and end of an expanding epidemic cannot be precisely forecast”. en. In: *Proceedings of the National Academy of Sciences* 117.42 (Oct. 20, 2020), pp. 26190–26196. ISSN: 0027-8424, 1091-6490. DOI: `10.1073/pnas.2007868117`.

[53] Tianfeng Chai and R. Draxler. “Root mean square error (RMSE) or mean absolute error (MAE)?” In: *Geosci. Model Dev.* 7 (Jan. 31, 2014). DOI: `10.5194/gmdd-7-1525-2014`.

[54] Sourav Chakraborty et al. “Reaction order and neural network approaches for the simulation of COVID-19 spreading kinetic in India”. en. In: *Infectious Disease Modelling* 5 (Jan. 1, 2020), pp. 737–747. ISSN: 2468-0427. DOI: `10.1016/j.idm.2020.09.002`.

[55] Satish Chander, Vijaya Padmanabha, and Joseph Mani. “Jaya Spider Monkey Optimization-driven Deep Convolutional LSTM for the prediction of COVID’19”. In: *Bio-Algorithms and Med-Systems* 16 (Nov. 13, 2020). DOI: `10.1515/bams-2020-0030`.

[56] Ahan Chatterjee and Swagatam Roy. "An Analytics Overview & LSTM-Based Predictive Modeling of Covid-19: A Hardheaded Look Across India". In: *Machine Intelligence and Soft Computing*. Ed. by Debnath Bhattacharyya and N. Thirupathi Rao. Singapore: Springer Singapore, 2021, pp. 289–307. ISBN: 978-981-15-9516-5.

[57] Ayan Chatterjee, Martin W. Gerdes, and Santiago G. Martinez. "Statistical Explorations and Univariate Timeseries Analysis on COVID-19 Datasets to Understand the Trend of Disease Spreading and Death". In: *Sensors* 20.11 (May 29, 2020). ISSN: 1424-8220. DOI: 10.3390/s20113089.

[58] Chen Chen. "Analysis and Forecast of Traffic Accident Big Data". In: *ITM Web of Conferences* 12 (2017), p. 04029. ISSN: 2271-2097. DOI: 10.1051/itmconf/20171204029.

[59] Jianguo Chen et al. "A Survey on Applications of Artificial Intelligence in Fighting Against COVID-19". In: *arXiv:2007.02202 [cs, q-bio]* (July 4, 2020).

[60] Jonathan H. Chen and Steven M. Asch. "Machine Learning and Prediction in Medicine — Beyond the Peak of Inflated Expectations". In: *The New England journal of medicine* 376.26 (June 29, 2017), pp. 2507–2509. ISSN: 0028-4793. DOI: 10.1056/NEJMp1702071.

[61] Li-Pang Chen. *Analysis and Prediction of COVID-19 Data in Taiwan*. en. Tech. rep. Rochester, NY: Social Science Research Network, May 27, 2020. DOI: 10.2139/ssrn.3611761.

[62] Quanjun Chen et al. "Learning Deep Representation from Big and Heterogeneous Data for Traffic Accident Inference". In: *AAAI: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. Phoenix, USA*. 2016.

[63] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16* (2016), pp. 785–794. DOI: 10.1145/2939672.2939785. arXiv: 1603.02754.

[64] Yuanhang Chen et al. "A novel deep learning method based on attention mechanism for bearing remaining useful life prediction". en. In: *Applied Soft Computing* 86 (Jan. 2020), p. 105919. ISSN: 1568-4946. DOI: 10.1016/j.asoc.2019.105919.

[65] Weiyu Cheng et al. "A Neural Attention Model for Urban Air Quality Inference: Learning the Weights of Monitoring Stations". In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. Ed. by Sheila A. McIlraith and Kilian Q. Weinberger. AAAI Press, 2018, pp. 2151–2158.

[66] Vinay Kumar Reddy Chimmula and Lei Zhang. "Time series forecasting of COVID-19 transmission in Canada using LSTM networks". en. In: *Chaos, Solitons & Fractals* 135 (June 1, 2020), p. 109864. ISSN: 0960-0779. DOI: 10.1016/j.chaos.2020.109864.

[67] Kyunghyun Cho et al. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". In: *arXiv:1406.1078 [cs, stat]* (Sept. 2, 2014).

[68] Kyunghyun Cho et al. "On the Properties of Neural Machine Translation: Encoder–Decoder Approaches". In: *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 103–111. DOI: 10.3115/v1/W14-4012.

[69] Ahlame Douzal Chouakria and Panduranga Naidu Nagabhushan. "Adaptive dissimilarity index for measuring time series proximity". en. In: *Advances in Data Analysis and Classification* 1.1 (Feb. 2007), pp. 5–21. ISSN: 1862-5347, 1862-5355. DOI: 10.1007/s11634-006-0004-6.

[70] Zhicheng Cui, Wenlin Chen, and Yixin Chen. "Multi-Scale Convolutional Neural Networks for Time Series Classification". In: *arXiv:1603.06995 [cs]* (May 2016).

[71] Zhiyong Cui et al. "High-Order Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting". In: *Transportation Research Board 98th Annual Meeting*. 2019.

[72] Zhiyong Cui et al. "Traffic Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting". In: *IEEE Transactions on Intelligent Transportation Systems* (2019), pp. 1–12. ISSN: 1558-0016. DOI: 10.1109/TITS.2019.2950416.

[73] Martin Cunneen et al. "Artificial Driving Intelligence and Moral Agency: Examining the Decision Ontology of Unavoidable Road Traffic Accidents through the Prism of the Trolley Dilemma". In: *Applied Artificial Intelligence* 33.3 (2019), pp. 267–293. ISSN: 0883-9514. DOI: 10.1080/08839514.2018.1560124.

[74] B. Curry and P. H. Morgan. "Neural networks, linear functions and neglected non-linearity". en. In: *Computational Management Science* 1.1 (Dec. 1, 2003), pp. 15–29. ISSN: 1619-6988. DOI: 10.1007/s10287-003-0003-4.

[75] Xingyuan Dai et al. "DeepTrend 2.0: A light-weighted multi-scale traffic prediction model using detrending". In: *Transportation Research Part C: Emerging Technologies* 103 (June 2019), pp. 142–157. ISSN: 0968-090X. DOI: 10.1016/j.trc.2019.03.022.

[76] Raj Dandekar and George Barbastathis. "Neural Network aided quarantine control model estimation of global Covid-19 spread". In: *arXiv:2004.02752 [physics, q-bio]* (Apr. 2, 2020).

[77] Andrey Davydenko and Robert Fildes. *Forecast Error Measures: Critical Review and Practical Recommendations*. Jan. 2016. DOI: 10.13140/RG.2.1.4539.5281.

[78] Edouard Delasalles et al. "Spatio-temporal neural networks for space-time data modeling and relation discovery". en. In: *Knowledge and Information Systems* 61.3 (Dec. 2019), pp. 1241–1267. ISSN: 0219-3116. DOI: 10.1007/s10115-018-1291-x.

[79] Shaojiang Deng, Shuyuan Jia, and Jing Chen. "Exploring spatial–temporal relations via deep convolutional neural networks for traffic flow prediction with incomplete data". en. In: *Applied Soft Computing* 78 (May 2019), pp. 712–721. ISSN: 1568-4946. DOI: 10.1016/j.asoc.2018.09.040.

[80] Jayanthi Devaraj et al. "Forecasting of COVID-19 cases using Deep learning models: Is it reliable and practically significant?" en. In: *Results in Physics* (Jan. 14, 2021), p. 103817. ISSN: 2211-3797. DOI: 10.1016/j.rinp.2021.103817.

[81] S. Dhamodharavadhani, R. Rathipriya, and Jyotir Moy Chatterjee. "COVID-19 Mortality Rate Prediction for India Using Statistical Neural Network Models". English. In: *Frontiers in Public Health* 8 (2020). ISSN: 2296-2565. DOI: 10.3389/fpubh.2020.00441.

[82] *Digital Earth*. en-US. URL: https://www.digitalearth.art.

[83] Cem Direkoglu and Melike Sah. "Worldwide and Regional Forecasting of Coronavirus (Covid-19) Spread using a Deep Learning Model". en. In: *medRxiv* (May 26, 2020), p. 2020.05.23.20111039. DOI: 10.1101/2020.05.23.20111039.

[84] Cosimo Distante et al. "Forecasting Covid-19 Outbreak Progression in Italian Regions: A model based on neural network training from Chinese data". en. In: *medRxiv* (Apr. 14, 2020), p. 2020.04.09.20059055. DOI: 10.1101/2020.04.09.20059055.

[85] Loan N. N. Do et al. "An effective spatial-temporal attention based neural network for traffic flow prediction". en. In: *Transportation Research Part C: Emerging Technologies* 108 (Nov. 2019), pp. 12–28. ISSN: 0968-090X. DOI: 10.1016/j.trc.2019.09.008.

[86] Xuchen Dong et al. "Short-Term Traffic Flow Prediction Based on XGBoost". In: *2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS)*. Enshi: IEEE, May 2018, pp. 854–859. ISBN: 978-1-5386-2618-4. DOI: 10.1109/DDCLS.2018.8516114.

[87] Jiaoman Du et al. "Urban hazmat transportation with multi-factor". en. In: *Soft Computing* (2019). ISSN: 1433-7479. DOI: 10.1007/s00500-019-03956-x.

[88] Hrishikesh Dutta. "Neural Network Model for Prediction of Covid-19 Confirmed Cases and Fatalities". In: *Preprint* (May 1, 2020).

[89] Shawni Dutta and Samir Kumar Bandyopadhyay. "Machine Learning Approach for Confirmation of COVID-19 Cases: Positive, Negative, Death and Release". en. In: *medRxiv* (Mar. 30, 2020), p. 2020.03.25.20043505. DOI: 10.1101/2020.03.25.20043505.

[90] Julio Díaz and Cristina Linares. *Informe sobre sostenibilidad en España 2019: por qué las ciudades son clave en la transición ecológica*. es. Madrid: Fundación Alternativas, 2019. ISBN: 978-84-12-02483-8.

[91] Effati Meysam et al. "Prediction of Crash Severity on Two-Lane, Two-Way Roads Based on Fuzzy Classification and Regression Tree Using Geospatial Analysis". In: *Journal of Computing in Civil Engineering* 29.6 (Nov. 2015), p. 04014099. DOI: 10.1061/(ASCE)CP.1943-5487.0000432.

[92] Ammar H. Elsheikh et al. "Deep learning-based forecasting model for COVID-19 outbreak in Saudi Arabia". en. In: *Process Safety and Environmental Protection* 149 (May 1, 2021), pp. 223–233. ISSN: 0957-5820. DOI: 10.1016/j.psep.2020.10.048.

[93] Alireza Ermagun and David Levinson. "Spatiotemporal traffic forecasting: review and proposed directions". en. In: *Transport Reviews* 38.6 (Nov. 2018), pp. 786–814. ISSN: 0144-1647, 1464-5327. DOI: 10.1080/01441647.2018.1442887.

[94] Reza Eshtehadi, Emrah Demir, and Yuan Huang. "Solving the vehicle routing problem with multi-compartment vehicles for city logistics". en. In: *Computers & Operations Research* 115 (Mar. 2020), p. 104859. ISSN: 0305-0548. DOI: 10.1016/j.cor.2019.104859.

[95] Daniel Fink. "A Compendium of Conjugate Priors". In: *Environmental Statistical group, Department of Biology, Montana State University, USA* (1997).

[96] A. S. Fokas, N. Dikaios, and G. A. Kastis. "Mathematical models and deep learning for predicting the number of individuals reported to be infected with SARS-CoV-2". In: *Journal of The Royal Society Interface* 17.169 (2020), p. 20200494. DOI: 10.1098/rsif.2020.0494.

[97] Simon James Fong et al. "Composite Monte Carlo decision making under high uncertainty of novel coronavirus epidemic using hybridized deep learning and fuzzy rule induction". en. In: *Applied Soft Computing* 93 (Aug. 1, 2020), p. 106282. ISSN: 1568-4946. DOI: 10.1016/j.asoc.2020.106282.

[98] Simon James Fong et al. "Finding an Accurate Early Forecasting Model from Small Dataset: A Case of 2019-nCoV Novel Coronavirus Outbreak". en. In: *International Journal of Interactive Multimedia and Artificial Intelligence* 6.Special Issue on Soft Computing (2020). ISSN: 1989-1660.

[99] Xingbo Fu et al. "Spatiotemporal Attention Networks for Wind Power Forecasting". In: *arXiv:1909.07369 [cs]* (Sept. 2019).

[100] Fabio Galatioto et al. “Advanced accident prediction models and impacts assessment”. en. In: *IET Intelligent Transport Systems* 12.9 (Nov. 2018), pp. 1131–1141. ISSN: 1751-956X, 1751-9578. DOI: 10.1049/iet-its.2018.5218.

[101] *Galileo is the European global satellite-based navigation system | European Global Navigation Satellite Systems Agency*. URL: https://www.gsa.europa.eu/european-gnss/galileo/galileo-european-global-satellite-based-navigation-system.

[102] Yogesh Gautam. “Transfer Learning for COVID-19 cases and deaths forecast using LSTM network”. en. In: *ISA Transactions* (Jan. 4, 2021). ISSN: 0019-0578. DOI: 10.1016/j.isatra.2020.12.057.

[103] Nouby M. Ghazaly, Muhammad A. Abdel-Fattah, and A. A. Abd El-Aziz. “Novel Coronavirus Forecasting Model using Nonlinear Autoregressive Artificial Neural Network”. en. In: *International Journal of Advanced Science and Technology* 29.5s (Apr. 9, 2020), pp. 1831–1849. ISSN: 2005-4238.

[104] Tilmann Gneiting and Matthias Katzfuss. “Probabilistic Forecasting”. In: *Annual Review of Statistics and Its Application* 1.1 (2014), pp. 125–151. DOI: 10.1146/annurev-statistics-062713-085831. URL: https://doi.org/10.1146/annurev-statistics-062713-085831.

[105] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016. ISBN: 978-0-262-03561-3.

[106] C. W. J. Granger and Roselyne Joyeux. “An Introduction to Long-Memory Time Series Models and Fractional Differencing”. en. In: *Journal of Time Series Analysis* 1.1 (1980), pp. 15–29. ISSN: 1467-9892. DOI: https://doi.org/10.1111/j.1467-9892.1980.tb00297.x.

[107] G. Grivas and A. Chaloulakou. “Artificial neural network models for prediction of PM10 hourly concentrations, in the Greater Area of Athens, Greece”. en. In: *Atmospheric Environment* 40.7 (Mar. 2006), pp. 1216–1229. ISSN: 1352-2310. DOI: 10.1016/j.atmosenv.2005.10.036.

[108] Shengnan Guo et al. “Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting”. en. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01 (July 2019), pp. 922–929. ISSN: 2374-3468. DOI: 10.1609/aaai.v33i01.3301922.

[109] Shengnan Guo et al. “Deep Spatial–Temporal 3D Convolutional Neural Networks for Traffic Data Forecasting”. In: *IEEE Transactions on Intelligent Transportation Systems* 20.10 (Oct. 2019), pp. 3913–3926. ISSN: 1558-0016. DOI: 10.1109/TITS.2019.2906365.

[110] Hamed Mohammad M., Al-Masaeid Hashem R., and Said Zahi M. Bani. “Short-Term Prediction of Traffic Volume in Urban Arterials”. In: *Journal of Transportation Engineering* 121.3 (May 1995), pp. 249–254. DOI: 10.1061/(ASCE)0733-947X(1995)121:3(249).

[111] Guy S. Handelman et al. “Peering Into the Black Box of Artificial Intelligence: Evaluation Metrics of Machine Learning Methods”. In: *American Journal of Roentgenology* 212.1 (Oct. 17, 2018), pp. 38–43. ISSN: 0361-803X. DOI: 10.2214/AJR.18.20224.

[112] Pitoyo Hartono. “Similarity maps and pairwise predictions for transmission dynamics of COVID-19 with neural networks”. en. In: *Informatics in Medicine Unlocked* 20 (Jan. 1, 2020), p. 100386. ISSN: 2352-9148. DOI: 10.1016/j.imu.2020.100386.

[113] Najmul Hasan. “A Methodological Approach for Predicting COVID-19 Epidemic Using EEMD-ANN Hybrid Model”. en. In: *Internet of Things* 11 (Sept. 1, 2020), p. 100228. ISSN: 2542-6605. DOI: 10.1016/j.iot.2020.100228.

[114] S. Hassanzadeh, F. Hosseinibalam, and R. Alizadeh. “Statistical models and time series forecasting of sulfur dioxide: a case study Tehran”. en. In: *Environmental Monitoring and Assessment* 155.1-4 (Aug. 2009), pp. 149–155. ISSN: 0167-6369, 1573-2959. DOI: 10.1007/s10661-008-0424-1.

[115] Mohamed Hawas. “Generated time-series prediction data of COVID-19’s daily infections in Brazil by using recurrent neural networks”. en. In: *Data in Brief* 32 (Oct. 1, 2020), p. 106175. ISSN: 2352-3409. DOI: 10.1016/j.dib.2020.106175.

[116] Zhixiang He, Chi-Yin Chow, and Jia-Dong Zhang. “STCNN: A Spatio-Temporal Convolutional Neural Network for Long-Term Traffic Prediction”. In: *2019 20th IEEE International Conference on Mobile Data Management (MDM)*. ISSN: 1551-6245. June 2019. DOI: 10.1109/MDM.2019.00-53.

[117] G. E. Hinton and R. R. Salakhutdinov. “Reducing the Dimensionality of Data with Neural Networks”. en. In: *Science* 313.5786 (July 28, 2006), pp. 504–507. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1127647.

[118] Mun Chon Ho et al. “An improved pheromone-based vehicle rerouting system to reduce traffic congestion”. en. In: *Applied Soft Computing* 84 (Nov. 2019), p. 105702. ISSN: 1568-4946. DOI: 10.1016/j.asoc.2019.105702.

[119] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[120] J. Hoel et al. “Different forms of attentional disturbances involved in driving accidents”. en. In: *IET Intelligent Transport Systems* 5.2 (2011), p. 120. ISSN: 1751956X. DOI: 10.1049/iet-its.2010.0109.

[121] D. W. Hosmer et al. “A Comparison of Goodness-of-Fit Tests for the Logistic Regression Model”. en. In: *Statistics in Medicine* 16.9 (1997), pp. 965–980. ISSN: 1097-0258. DOI: https://doi.org/10.1002/(SICI)1097-0258(19970515)16:9{\textless}965::AID-SIM509{\textgreater}3.0.CO;2-O.

[122] Zixin Hu et al. "Artificial Intelligence Forecasting of Covid-19 in China". en. In: *Preprint* (Feb. 17, 2020). URL: `https://arxiv.org/abs/2002.07112v2`.

[123] Zixin Hu et al. "Forecasting and evaluating intervention of Covid-19 in the World". In: *arXiv:2003.09800 [q-bio]* (Mar. 21, 2020).

[124] Chiou-Jye Huang et al. "Multiple-Input Deep Convolutional Neural Network Model for COVID-19 Forecasting in China". en. In: *medRxiv* (Mar. 27, 2020), p. 2020.03.23.20041608. ISSN: 2004-1608. DOI: `10.1101/2020.03.23.20041608`.

[125] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. "Extreme learning machine: a new learning scheme of feedforward neural networks". In: *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*. Vol. 2. 2004, 985–990 vol.2. DOI: `10.1109/IJCNN.2004.1380068`.

[126] R.J. Hyndman and G. Athanasopoulos. *Forecasting: Principles and Practice (3rd ed)*. 3rd ed. Melbourne, Australia: OTexts, 2021.

[127] Rob J Hyndman and Yeasmin Khandakar. "Automatic time series forecasting: the forecast package for R". In: *Journal of Statistical Software* 26.3 (2008), pp. 1–22. URL: `https://www.jstatsoft.org/article/view/v027i03`.

[128] Marie-Eve Héroux et al. "Quantifying the health impacts of ambient air pollutants: recommendations of a WHO/Europe project". en. In: *International Journal of Public Health* 60.5 (July 2015), pp. 619–627. ISSN: 1661-8564. DOI: `10.1007/s00038-015-0690-y`.

[129] Mohamed R. Ibrahim et al. "Variational-LSTM Autoencoder to forecast the spread of coronavirus across the globe". en. In: *medRxiv* (Apr. 24, 2020), p. 2020.04.20.20070938. DOI: `10.1101/2020.04.20.20070938`.

[130] Othman Istaiteh et al. "Machine Learning Approaches for COVID-19 Forecasting". In: *2020 International Conference on Intelligent Data Science Technologies and Applications (IDSTA)*. 2020, pp. 50–57. DOI: `10.1109/IDSTA50958.2020.9264101`.

[131] Alireza M. Javid et al. "Predictive Analysis of COVID-19 Time-series Data from Johns Hopkins University". In: *arXiv:2005.05060 [cs, eess]* (May 22, 2020).

[132] Dohyoung Jo et al. "Image-to-Image Learning to Predict Traffic Speeds by Considering Area-Wide Spatio-Temporal Dependencies". In: *IEEE Transactions on Vehicular Technology* 68.2 (Feb. 2019), pp. 1188–1197. ISSN: 1939-9359. DOI: `10.1109/TVT.2018.2885366`.

[133] Frank E. Harrell Jr. *Hmisc: Harrell Miscellaneous*. Aug. 2020. URL: `https://CRAN.R-project.org/package=Hmisc`.

[134] Rahele Kafieh et al. "COVID-19 in Iran: A Deeper Look Into The Future". en. In: *medRxiv* (Apr. 27, 2020), p. 2020.04.24.20078477. DOI: `10.1101/2020.04.24.20078477`.

[135] Egide Kalisa et al. “Temperature and air pollution relationship during heatwaves in Birmingham, UK”. en. In: *Sustainable Cities and Society* 43 (Nov. 2018), pp. 111–120. ISSN: 2210-6707. DOI: 10.1016/j.scs.2018.08.033.

[136] Amol Kapoor et al. “Examining COVID-19 Forecasting using Spatio-Temporal Graph Neural Networks”. In: *arXiv:2007.03113 [cs]* (July 6, 2020).

[137] Armin Kappeler et al. “Video Super-Resolution With Convolutional Neural Networks”. In: *IEEE Transactions on Computational Imaging* 2.2 (June 2016), pp. 109–122. ISSN: 2333-9403. DOI: 10.1109/TCI.2016.2532323.

[138] Y. Karadayi, M. N. Aydin, and A. S. Öğrencí. “Unsupervised Anomaly Detection in Multivariate Spatio-Temporal Data Using Deep Learning: Early Detection of COVID-19 Outbreak in Italy”. In: *IEEE Access* 8 (2020), pp. 164155–164177. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3022366.

[139] Md Karimuzzaman et al. “Forecasting the COVID-19 Pandemic with Climate Variables for Top Five Burdening and Three South Asian Countries”. en. In: *medRxiv* (May 19, 2020), p. 2020.05.12.20099044. DOI: 10.1101/2020.05.12.20099044.

[140] Jintao Ke et al. “Hexagon-Based Convolutional Neural Network for Supply-Demand Forecasting of Ride-Sourcing Services”. In: *IEEE Transactions on Intelligent Transportation Systems* 20.11 (Nov. 2019), pp. 4160–4173. ISSN: 1558-0016. DOI: 10.1109/TITS.2018.2882861.

[141] Frank J. Kelly and Julia C. Fussell. “Air pollution and public health: emerging hazards and improved understanding of risk”. In: *Environmental Geochemistry and Health* 37.4 (2015), pp. 631–649.

[142] Sultan Daud Khan, Louai Alarabi, and Saleh Basalamah. “Toward Smart Lockdown: A Novel Approach for COVID-19 Hotspots Prediction Using a Deep Hybrid Neural Network”. en. In: *Computers* 9.4 (Dec. 11, 2020), p. 99. ISSN: 2073-431X. DOI: 10.3390/computers9040099.

[143] Ki-Hyun Kim, Ehsanul Kabir, and Shamin Kabir. “A review on the human health impact of airborne particulate matter”. en. In: *Environment International* 74 (Jan. 2015), pp. 136–143. ISSN: 0160-4120. DOI: 10.1016/j.envint.2014.10.005.

[144] Kyung Hwan Kim et al. “Influence of wind direction and speed on the transport of particle-bound PAHs in a roadway environment”. en. In: *Atmospheric Pollution Research* 6.6 (Nov. 2015), pp. 1024–1034. ISSN: 1309-1042. DOI: 10.1016/j.apr.2015.05.007.

[145] Roger Koenker. *Quantile Regression*. Econometric Society Monographs. Cambridge: Cambridge University Press, 2005. ISBN: 978-0-521-84573-1. DOI: 10.1017/CBO9780511754098.

[146] Laszlo Robert Kolozsvari et al. "Predicting the epidemic curve of the coronavirus (SARS-CoV-2) disease (COVID-19) using artificial intelligence". en. In: *medRxiv* (Jan. 27, 2021), p. 2020.04.17.20069666. DOI: 10.1101/2020.04.17.20069666.

[147] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira et al. Curran Associates, Inc., 2012, pp. 1097–1105.

[148] Aishwarya Kumar, Puneet Kumar Gupta, and Ankita Srivastava. "A review of modern technologies for tackling COVID-19 pandemic". eng. In: *DiabetesMetabolic Syndrome* 14.4 (Aug. 2020), pp. 569–573. ISSN: 1878-0334. DOI: 10.1016/j.dsx.2020.05.008.

[149] S. Vasantha Kumar and Lelitha Vanajakshi. "Short-term traffic flow prediction using seasonal ARIMA model with limited input data". en. In: *European Transport Research Review* 7.3 (June 2015), p. 21. ISSN: 1866-8887. DOI: 10.1007/s12544-015-0170-8.

[150] Ujjwal Kumar and V. K. Jain. "ARIMA forecasting of ambient air pollutants (O3, NO, NO2 and CO)". en. In: *Stochastic Environmental Research and Risk Assessment* 24.5 (July 2010), pp. 751–760. ISSN: 1436-3259. DOI: 10.1007/s00477-009-0361-8.

[151] İsmail Kırbaş et al. "Comparative analysis and forecasting of COVID-19 cases in various European countries with ARIMA, NARNN and LSTM approaches". en. In: *Chaos, Solitons & Fractals* 138 (Sept. 1, 2020), p. 110015. ISSN: 0960-0779. DOI: 10.1016/j.chaos.2020.110015.

[152] Guokun Lai et al. "Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks". In: *arXiv:1703.07015 [cs]* (Apr. 2018).

[153] Samuel Lalmuanawma, Jamal Hussain, and Lalrinfela Chhakchhuak. "Applications of machine learning and artificial intelligence for Covid-19 (SARS-CoV-2) pandemic: A review". en. In: *Chaos, Solitons & Fractals* 139 (Oct. 1, 2020), p. 110059. ISSN: 0960-0779. DOI: 10.1016/j.chaos.2020.110059.

[154] S. Latif et al. "Leveraging Data Science to Combat COVID-19: A Comprehensive Review". In: *IEEE Transactions on Artificial Intelligence* 1.1 (Aug. 2020), pp. 85–103. ISSN: 2691-4581. DOI: 10.1109/TAI.2020.3020521.

[155] Charles L Lawson and Richard J Hanson. *Solving least squares problems*. SIAM, 1995.

[156] D. Lazer et al. "The Parable of Google Flu: Traps in Big Data Analysis". en. In: *Science* 343.6176 (Mar. 14, 2014), pp. 1203–1205. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1248506.

[157] Yann Lecun. "Generalization and network design strategies". English (US). In: *Connectionism in perspective* (1989). URL: https://nyuscholars.nyu.edu/en/publications/generalization-and-network-design-strategies.

[158] Yann Lecun. *PhD thesis: Modeles connexionnistes de l'apprentissage (connectionist learning models)*. English (US). Universite P. et M. Curie (Paris 6), June 1987.

[159] Yann LeCun et al. "Backpropagation applied to handwritten zip code recognition". In: *Neural computation* 1.4 (1989), pp. 541–551.

[160] Sang-Il Lee. "Correlation and Spatial Autocorrelation". In: *Encyclopedia of GIS*. Ed. by Shashi Shekhar, Hui Xiong, and Xun Zhou. Cham: Springer International Publishing, 2017, pp. 360–368. ISBN: 978-3-319-17885-1. DOI: 10.1007/978-3-319-17885-1_1524.

[161] Yee Leung et al. "Integration of air pollution data collected by mobile sensors and ground-based stations to derive a spatiotemporal air pollution profile of a city". en. In: *International Journal of Geographical Information Science* 33.11 (Nov. 2019), pp. 2218–2240. ISSN: 1365-8816, 1362-3087. DOI: 10.1080/13658816.2019.1633468.

[162] Oscar Li et al. "Deep Learning for Case-Based Reasoning through Prototypes: A Neural Network that Explains Its Predictions". en. In: *The Thirty-Second AAAI Conference on Artificial Intelligence* (2018).

[163] Xiugang Li et al. "Predicting motor vehicle crashes using Support Vector Machine models". In: *Accident Analysis & Prevention* 40.4 (2008), pp. 1611–1618. ISSN: 0001-4575. DOI: 10.1016/j.aap.2008.04.010.

[164] Yaguang Li et al. "Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting". In: *arXiv:1707.01926 [cs, stat]* (Feb. 2018).

[165] Zhiheng Li, Yuebiao Li, and Li Li. "A comparison of detrending models and multi-regime models for traffic flow prediction". In: *IEEE Intelligent Transportation Systems Magazine* 6.4 (2014), pp. 34–44.

[166] Zhijian Li et al. "A Recurrent Neural Network and Differential Equation Based Spatiotemporal Infectious Disease Model with Application to COVID-19". In: *arXiv:2007.10929 [cs, q-bio, stat]* (Sept. 17, 2020).

[167] Binbing Liao et al. "Deep Sequence Learning with Auxiliary Information for Traffic Prediction". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 2018.

[168] Binbing Liao et al. "Deep Sequence Learning with Auxiliary Information for Traffic Prediction". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '18. London, United Kingdom: Association for Computing Machinery, July 2018, pp. 537–546. ISBN: 978-1-4503-5552-0. DOI: 10.1145/3219819.3219895.

[169] Binbing Liao et al. “Dest-ResNet: A Deep Spatiotemporal Residual Network for Hotspot Traffic Speed Prediction”. en. In: *2018 ACM Multimedia Conference on Multimedia Conference - MM '18*. Seoul, Republic of Korea: ACM Press, 2018, pp. 1883–1891. ISBN: 978-1-4503-5665-7. DOI: `10.1145/3240508.3240656`.

[170] Lei Lin, Qian Wang, and Adel W. Sadek. “A novel variable selection method based on frequent pattern tree for real-time traffic accident risk prediction”. In: *Transportation Research Part C: Emerging Technologies*. Engineering and Applied Sciences Optimization (OPT-i) - Professor Matthew G. Karlaftis Memorial Issue 55 (2015), pp. 444–459. ISSN: 0968-090X. DOI: `10.1016/j.trc.2015.03.015`.

[171] T. A. Litman. “Transportation Cost and Benefit Analysis: Techniques, Estimates and Implications”. In: *Victoria Transport Policy Institute, 2nd ed.* (2009), pp. 1–19.

[172] Boyi Liu et al. “Traffic Flow Combination Forecasting Method Based on Improved LSTM and ARIMA”. en. In: *International Journal of Embedded Systems* 12.1 (2020), pp. 22 –30.

[173] Yi Liu et al. “Scalable privacy-enhanced traffic monitoring in vehicular ad hoc networks”. en. In: *Soft Computing* 20.8 (2016), pp. 3335–3346. ISSN: 1433-7479. DOI: `10.1007/s00500-015-1737-y`.

[174] Z. Liu et al. “Urban Traffic Prediction from Mobility Data Using Deep Learning”. In: *IEEE Network* 32.4 (July 2018), pp. 40–46. ISSN: 0890-8044. DOI: `10.1109/MNET.2018.1700411`.

[175] Zhenbing Liu et al. “Spatiotemporal saliency-based multi-stream networks with attention-aware LSTM for action recognition”. en. In: *Neural Computing and Applications* 32.18 (Sept. 2020), pp. 14593–14602. ISSN: 1433-3058. DOI: `10.1007/s00521-020-05144-7`.

[176] Dominique Lord. “Modeling motor vehicle crashes using Poisson-gamma models: Examining the effects of low sample mean values and small sample size on the estimation of the fixed dispersion parameter”. In: *Accident Analysis & Prevention* 38.4 (2006), pp. 751 –766. ISSN: 0001-4575. DOI: `https://doi.org/10.1016/j.aap.2006.02.001`.

[177] Dominique Lord and Fred Mannering. “The statistical analysis of crash-frequency data: A review and assessment of methodological alternatives”. In: *Transportation Research Part A: Policy and Practice* 44.5 (2010), pp. 291–305. ISSN: 0965-8564. DOI: `10.1016/j.tra.2010.02.001`.

[178] Feng Lu et al. “Modeling the heterogeneous traffic correlations in urban road systems using traffic-enhanced community detection approach”. en. In: *Physica A: Statistical Mechanics and its Applications* 501 (July 2018), pp. 227–237. ISSN: 0378-4371. DOI: `10.1016/j.physa.2018.02.062`.

[179] Scott M Lundberg and Su-In Lee. “A unified approach to interpreting model predictions”. In: *Advances in neural information processing systems*. 2017, pp. 4765–4774.

[180] Wei Luo et al. "Guidelines for Developing and Reporting Machine Learning Predictive Models in Biomedical Research: A Multidisciplinary View". EN. In: *Journal of Medical Internet Research* 18.12 (Dec. 16, 2016), pp. 58–70. DOI: 10.2196/jmir.5870.

[181] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. "Effective Approaches to Attention-based Neural Machine Translation". In: *arXiv:1508.04025 [cs]* (Sept. 2015).

[182] Kim Tien Ly. "A COVID-19 forecasting system using adaptive neuro-fuzzy inference". en. In: *Finance Research Letters* (Nov. 12, 2020), p. 101844. ISSN: 1544-6123. DOI: 10.1016/j.frl.2020.101844.

[183] Vít Macháček and Martin Srholec. "Predatory publishing in Scopus: evidence on cross-country differences". en. In: *Scientometrics* (Feb. 7, 2021). ISSN: 1588-2861. DOI: 10.1007/s11192-020-03852-4.

[184] Piotr S. Maciąg et al. "Air pollution prediction with clustering-based ensemble of evolving spiking neural networks and a case study for London area". en. In: *Environmental Modelling & Software* 118 (Aug. 2019), pp. 262–280. ISSN: 1364-8152. DOI: 10.1016/j.envsoft.2019.04.012.

[185] Madrid-Council. *En portada - Portal de datos abiertos del Ayuntamiento de Madrid*. es. URL: https://datos.madrid.es/portal/site/egob.

[186] Madrid-Protocol. *Protocolo de actuación para episodios de contaminación por dióxido de nitrógeno - Ayuntamiento de Madrid*. es. 2018. URL: http://www.mambiente.madrid.es/opencms/calaire.

[187] Rajvikram Madurai Elavarasan and Rishi Pugazhendhi. "Restructured society and environment: A review on potential technological strategies to control the COVID-19 pandemic". en. In: *Science of The Total Environment* 725 (July 10, 2020), p. 138858. ISSN: 0048-9697. DOI: 10.1016/j.scitotenv.2020.138858.

[188] Parikshit N. Mahalle et al. "Data Analytics: COVID-19 Prediction Using Multimodal Data". en. In: *Preprint* (May 14, 2020). DOI: 10.20944/preprints202004.0257.v2.

[189] Fred L. Mannering and Chandra R. Bhat. "Analytic methods in accident research: Methodological frontier and future directions". In: *Analytic Methods in Accident Research* 1 (2014), pp. 1–22. ISSN: 2213-6657. DOI: 10.1016/j.amar.2013.09.001.

[190] Joao Alexandre Lobo Marques et al. "Artificial Intelligence Prediction for the COVID-19 Data Based on LSTM Neural Networks and H2O AutoML". en. In: *Predictive Models for Decision Support in the COVID-19 Crisis*. SpringerBriefs in Applied Sciences and Technology. Cham: Springer International Publishing, 2021, pp. 69–87. ISBN: 978-3-030-61913-8.

[191] Marco Martuzzi et al. "Health impact of PM10 and ozone in 13 Italian cities". In: *WHO Regional Office for Europe* (2006), p. 133.

[192] V. Marécal et al. "A regional air quality forecasting system over Europe: the MACC-II daily ensemble production". English. In: *Geoscientific Model Development* 8.9 (Sept. 2015), pp. 2777–2813. ISSN: 1991-959X. DOI: `https://doi.org/10.5194/gmd-8-2777-2015`.

[193] Warren S. McCulloch and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity". en. In: *The bulletin of mathematical biophysics* 5.4 (Dec. 1, 1943), pp. 115–133. ISSN: 1522-9602. DOI: `10.1007/BF02478259`.

[194] Rodrigo de Medrano and José L Aznarte. "A New Spatio-Temporal Neural Network Approach for Traffic Accident Forecasting". In: *Applied Artificial Intelligence* (2021), pp. 1–20. DOI: `10.1080/08839514.2021.1935588`.

[195] Rodrigo de Medrano and José L Aznarte. "On the inclusion of spatial information for spatio-temporal neural networks". In: *Neural Computing and Applications* (2021), pp. 1–18. DOI: `10.1007/s00521-021-06111-6`.

[196] Rodrigo de Medrano and José L. Aznarte. "A spatio-temporal attention-based spot-forecasting framework for urban traffic prediction". en. In: *Applied Soft Computing* 96 (Nov. 2020), p. 106615. ISSN: 1568-4946. DOI: `10.1016/j.asoc.2020.106615`.

[197] Rodrigo de Medrano, Víctor de Buen Remiro, and José L Aznarte. "SOCAIRE: Forecasting and Monitoring Urban Air Quality in Madrid". In: *Environmental Modelling & Software* 143 (Sept. 2021), p. 105084. DOI: `10.1016/j.envsoft.2021.105084`.

[198] Patricia Melin et al. "Multiple Ensemble Neural Network Models with Fuzzy Response Aggregation for Predicting COVID-19 Time Series: The Case of Mexico". eng. In: *Healthcare (Basel, Switzerland)* 8.2 (June 19, 2020). ISSN: 2227-9032. DOI: `10.3390/healthcare8020181`.

[199] Zhang Mingheng et al. *Accurate Multisteps Traffic Flow Prediction Based on SVM*. en. Research article. 2013. DOI: `10.1155/2013/418303`.

[200] Leila Moftakhar, Mozhgan Seif, and Marziyeh Safe. "Exponentially Increasing Trend of Infected Patients with COVID-19 in Iran: A Comparison of Neural Network and ARIMA Forecasting Models". In: *Iranian Journal of Public Health* 49 (July 11, 2020). DOI: `10.18502/ijph.v49iS1.3675`.

[201] Youssoufa Mohamadou, Aminou Halidou, and Pascalin Tiam Kapen. "A review of mathematical modeling, artificial intelligence and datasets used in the study, prediction and management of COVID-19". en. In: *Applied Intelligence* (2020).

[202] Lucas Mohimont et al. "Convolutional Neural Networks and Temporal CNNs for Covid-19 Forecasting in France". en. In: *Applied Intelligence* (2021).

[203] Abolfazl Mollalo, Kiara M. Rivera, and Behzad Vahedi. "Artificial Neural Network Modeling of Novel Coronavirus (COVID-19) Incidence Rates across the Continental United States". en. In: *International Journal of Environmental Research and Public Health* 17.12 (Jan. 2020), p. 4204. DOI: 10.3390/ijerph17124204.

[204] John Mongan, Linda Moy, and Charles E. Kahn. In: *Radiology: Artificial Intelligence* 2.2 (Mar. 1, 2020), e200029. DOI: 10.1148/ryai.2020200029.

[205] P. A. P. Moran. "Notes on Continuous Stochastic Phenomena". In: *Biometrika* 37.1/2 (1950), pp. 17–23. ISSN: 00063444.

[206] Rachida Moulay Taj et al. "Towards Using Recurrent Neural Networks for Predicting Influenza-like Illness: Case Study of Covid-19 in Morocco". In: *International Journal of Advanced Trends in Computer Science and Engineering* 9 (Oct. 22, 2020), pp. 7945–7950. DOI: 10.30534/ijatcse/2020/148952020.

[207] Hyeonseob Nam and Bohyung Han. "Learning Multi-Domain Convolutional Neural Networks for Visual Tracking". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 4293–4302.

[208] Moncef Ilies Nasri, Tolga Bektaş, and Gilbert Laporte. "Route and speed optimization for autonomous trucks". en. In: *Computers & Operations Research* 100 (2018), pp. 89–101. ISSN: 0305-0548. DOI: 10.1016/j.cor.2018.07.015.

[209] Wim Naudé. *Artificial Intelligence Against Covid-19: An Early Review*. en. Tech. rep. Rochester, NY: Social Science Research Network, Apr. 6, 2020. URL: https://papers.ssrn.com/abstract=3568314.

[210] Wim Naudé. "Artificial intelligence vs COVID-19: limitations, constraints and pitfalls". en. In: *AISOCIETY* 35.3 (Sept. 1, 2020), pp. 761–765. ISSN: 1435-5655. DOI: 10.1007/s00146-020-00978-0.

[211] Ricardo Navares and José L. Aznarte. "Predicting air quality with deep learning LSTM: Towards comprehensive models". en. In: *Ecological Informatics* 55 (Jan. 2020), p. 101019. ISSN: 1574-9541. DOI: 10.1016/j.ecoinf.2019.101019.

[212] Asaf Nebenzal and Barak Fishbain. "Long-term forecasting of nitrogen dioxide ambient levels in metropolitan areas using the discrete-time Markov model". en. In: *Environmental Modelling & Software* 107 (Sept. 2018), pp. 175–185. ISSN: 1364-8152. DOI: 10.1016/j.envsoft.2018.06.001.

[213] Roger B. Nelsen. *An Introduction to Copulas*. en. Lecture Notes in Statistics. New York: Springer-Verlag, 1999. ISBN: 978-1-4757-3076-0. DOI: 10.1007/978-1-4757-3076-0. URL: https://www.springer.com/gp/book/9781475730760.

[214] Michael Nelson et al. “Time series forecasting using neural networks: should the data be deseasonalized first?” fr. In: *Journal of Forecasting* 18.5 (1999), pp. 359–367. ISSN: 1099-131X. DOI: 10.1002/(SICI)1099-131X(199909)18:5<359::AID-FOR746>3.0.CO;2-P.

[215] Dinh C. Nguyen et al. “Blockchain and AI-Based Solutions to Combat Coronavirus (COVID-19)-like Epidemics: A Survey”. en. In: *Preprint* (Apr. 19, 2020). DOI: 10.20944/preprints202004.0325.v1.

[216] Tu Nguyen. “Spatiotemporal Tile-based Attention-guided LSTMs for Traffic Video Prediction”. In: *arXiv:1910.11030 [cs, eess]* (Oct. 2019).

[217] Regina Nuzzo. “Scientific method: Statistical errors”. en. In: *Nature News* 506.7487 (Feb. 13, 2014), p. 150. DOI: 10.1038/506150a.

[218] Erick Meira de Oliveira and Fernando Luiz Cyrino Oliveira. “Forecasting mid-long term electric energy consumption through bagging ARIMA and exponential smoothing methods”. en. In: *Energy* 144 (Feb. 2018), pp. 776–788. ISSN: 0360-5442. DOI: 10.1016/j.energy.2017.12.049.

[219] Wei Ouyang et al. “The washing effect of precipitation on particulate matter and the pollution dynamics of rainwater in downtown Beijing”. en. In: *Science of The Total Environment* 505 (Feb. 2015), pp. 306–314. ISSN: 0048-9697. DOI: 10.1016/j.scitotenv.2014.09.062.

[220] Halûk Özkaynak et al. “Summary and findings of the EPA and CDC symposium on air pollution exposure and health”. In: *Journal of exposure science & environmental epidemiology* 19.1 (2009), pp. 19–29.

[221] Juan de Oña, Griselda López, and Joaquín Abellán. “Extracting decision rules from police accident reports through decision trees”. In: *Accident Analysis & Prevention* 50 (2013), pp. 1151–1160. ISSN: 0001-4575. DOI: 10.1016/j.aap.2012.09.006.

[222] Juan de Oña, Randa Oqab Mujalli, and Francisco J. Calvo. “Analysis of traffic accident injury severity on Spanish rural highways using Bayesian networks”. In: *Accident Analysis & Prevention* 43.1 (2011), pp. 402–411. ISSN: 0001-4575. DOI: 10.1016/j.aap.2010.09.010.

[223] Vasilis Papastefanopoulos, Pantelis Linardatos, and Sotiris Kotsiantis. “COVID-19: A Comparison of Time Series Methods to Forecast Percentage of Active Cases per Population”. en. In: *Applied Sciences* 10.11 (Jan. 2020), p. 3880. DOI: 10.3390/app10113880.

[224] Seong Ho Park and Kyunghwa Han. “Methodologic Guide for Evaluating Clinical Performance and Effect of Artificial Intelligence Technology for Medical Diagnosis and Prediction”. In: *Radiology* 286.3 (Jan. 8, 2018), pp. 800–809. ISSN: 0033-8419. DOI: 10.1148/radiol.2017171920.

[225] Seong Ho Park and Herbert Y. Kressel. “Connecting Technological Innovation in Artificial Intelligence to Real-world Medical Practice through Rigorous Clinical Validation: What Peer-reviewed Medical Journals Could Do”. eng. In: *Journal of Korean Medical Science* 33.22 (May 28, 2018), e152. ISSN: 1598-6357. DOI: 10.3346/jkms.2018.33.e152.

[226] A. Pascale and M. Nicoli. “Adaptative Bayesian Network For Traffic Flow Prediction”. In: *IEEE Statistical Signal Processing Workshop (SSP)* (2011).

[227] Josh Patterson and Adam Gibson. *Deep Learning*. en. O’Reilly Media, Inc., Aug. 2017. ISBN: 978-1-4919-1425-0. URL: https://www.oreilly.com/library/view/deep-learning/9781491924570/.

[228] Swarna kamal Paul, Saikat Jana, and Parama Bhaumik. “A multivariate spatiotemporal spread model of COVID-19 using ensemble of ConvLSTM networks”. In: *medRxiv* (2020). DOI: 10.1101/2020.04.17.20069898.

[229] M. Peden et al. “World Report on Road Traffic Injury Prevention”. In: (2004).

[230] Igor Pereira et al. “Forecasting Covid-19 Dynamics in Brazil: A Data Driven Approach”. In: *International Journal of Environmental Research and Public Health* 17 (July 15, 2020), p. 5115. DOI: 10.3390/ijerph17145115.

[231] Guilherme Perin and S. Picek. “On the Influence of Optimizers in Deep Learning-based Side-channel Analysis”. In: *IACR Cryptol. ePrint Arch.* (2020).

[232] Q. Pham et al. “Artificial Intelligence (AI) and Big Data for Coronavirus (COVID-19) Pandemic: A Survey on the State-of-the-Arts”. In: *IEEE Access* 8 (Apr. 21, 2020), pp. 130820–130839. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3009328.

[233] Gergo Pinter et al. “COVID-19 Pandemic Prediction for Hungary; A Hybrid Machine Learning Approach”. In: *Mathematics* 8.6 (2020). ISSN: 2227-7390. DOI: 10.3390/math8060890.

[234] Resa Septiani Pontoh et al. “Effectiveness of the public health measures to prevent the spread of COVID-19”. en. In: *Commun. Math. Biol. Neurosci.* 2020.0 (June 18, 2020), Article ID 31. ISSN: 2052-2541. DOI: 10.28919/cmbn/4711.

[235] Anupam Prakash et al. “Spread Peak Prediction of Covid-19 using ANN and Regression (Workshop Paper)”. In: *2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM)*. 2020, pp. 356–365. DOI: 10.1109/BigMM50055.2020.00062.

[236] Sikakollu Prasanth et al. “Forecasting spread of COVID-19 using google trends: A hybrid GWO-deep learning approach”. en. In: *Chaos, Solitons & Fractals* 142 (Jan. 1, 2021), p. 110336. ISSN: 0960-0779. DOI: 10.1016/j.chaos.2020.110336.

[237] James M. Provenzale and Robert J. Stanley. "A systematic guide to reviewing a manuscript". eng. In: *AJR. American journal of roentgenology* 185.4 (Oct. 2005), pp. 848–854. ISSN: 0361-803X. DOI: `10.2214/AJR.05.0782`.

[238] Narinder Singh Punn, Sanjay Kumar Sonbhadra, and Sonali Agarwal. "COVID-19 Epidemic Analysis using Machine Learning and Deep Learning Algorithms". en. In: *medRxiv* (June 1, 2020), p. 2020.04.08.20057679. ISSN: 2005-7679. DOI: `10.1101/2020.04.08.20057679`.

[239] Chenye Qiu et al. "A Multiobjective Particle Swarm Optimization-Based Partial Classification for Accident Severity Analysis". In: *Applied Artificial Intelligence* 28.6 (2014), pp. 555–576. ISSN: 0883-9514. DOI: `10.1080/08839514.2014.923166`.

[240] Licheng Qu et al. "Daily long-term traffic flow forecasting based on a deep neural network". en. In: *Expert Systems with Applications* 121 (May 2019), pp. 304–312. ISSN: 0957-4174. DOI: `10.1016/j.eswa.2018.12.031`.

[241] Catriona M. Queen and Casper J. Albers. "Intervention and Causality: Forecasting Traffic Flows Using a Dynamic Bayesian Network". In: *Journal of the American Statistical Association* 104.486 (June 2009), pp. 669–681. ISSN: 0162-1459. DOI: `10.1198/jasa.2009.0042`.

[242] A. Ramchandani, C. Fan, and A. Mostafavi. "DeepCOVIDNet: An Interpretable Deep Learning Model for Predictive Surveillance of COVID-19 Using Heterogeneous Features and Their Interactions". In: *IEEE Access* 8 (2020), pp. 159915–159930. ISSN: 2169-3536. DOI: `10.1109/ACCESS.2020.3019989`.

[243] Jawad Rasheed et al. "A survey on artificial intelligence approaches in supporting frontline workers and decision makers for the COVID-19 pandemic". en. In: *Chaos, Solitons & Fractals* 141 (Dec. 1, 2020), p. 110337. ISSN: 0960-0779. DOI: `10.1016/j.chaos.2020.110337`.

[244] Hafiz Tayyab Rauf et al. "Time series forecasting of COVID-19 transmission in Asia Pacific countries using deep neural networks". en. In: *Personal and Ubiquitous Computing* (Jan. 10, 2021). ISSN: 1617-4917. DOI: `10.1007/s00779-020-01494-0`.

[245] Honglei Ren et al. "A Deep Learning Approach to the Citywide Traffic Accident Risk Prediction". In: *2018 IEEE International Conference on Intelligent Transportation Systems (ITSC)*. 2017. arXiv: `1710.09543`.

[246] *Retracted coronavirus (COVID-19) papers*. en-US. Apr. 29, 2020.

[247] Kyoung-Ah Rhee et al. "Spatial regression analysis of traffic crashes in Seoul". In: *Accident Analysis & Prevention* 91 (2016), pp. 190–199. ISSN: 0001-4575. DOI: `10.1016/j.aap.2016.02.023`.

[248] Rizk M. Rizk-Allah and Aboul Ella Hassanien. "COVID-19 forecasting based on an improved interior search algorithm and multi-layer feed forward neural network". In: *arXiv:2004.05960 [cs, eess]* (Apr. 6, 2020).

[249] Michael Roberts et al. "Common pitfalls and recommendations for using machine learning to detect and prognosticate for COVID-19 using chest radiographs and CT scans". en. In: *Nature Machine Intelligence* 3.3 (Mar. 2021), pp. 199–217. ISSN: 2522-5839. DOI: 10.1038/s42256-021-00307-0.

[250] Filipe Rodrigues and Francisco C. Pereira. "Beyond Expectation: Deep Joint Mean and Quantile Regression for Spatiotemporal Problems". In: *IEEE Transactions on Neural Networks and Learning Systems* (2020), pp. 1–13. ISSN: 2162-2388. DOI: 10.1109/TNNLS.2020.2966745.

[251] Alexander Rodriguez et al. "DeepCOVID: An Operational Deep Learning-driven Framework for Explainable Real-time COVID-19 Forecasting". en. In: *medRxiv* (Sept. 29, 2020), p. 2020.09.28.20203109. DOI: 10.1101/2020.09.28.20203109.

[252] F. Rosenblatt. "PRINCIPLES OF NEURODYNAMICS. PERCEPTRONS AND THE THEORY OF BRAIN MECHANISMS". In: *The American Journal of Psychology* 76.4 (1963), pp. 705–707. DOI: 10.2307/1419730.

[253] F. Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain". eng. In: *Psychological Review* 65.6 (Nov. 1958), pp. 386–408. ISSN: 0033-295X. DOI: 10.1037/h0042519.

[254] Arash M. Roshandeh, Bismark R. D. K. Agbelie, and Yongdoo Lee. "Statistical modeling of total crash frequency at highway intersections". In: *Journal of Traffic and Transportation Engineering (English Edition)* 3.2 (2016), pp. 166–171. ISSN: 2095-7564. DOI: 10.1016/j.jtte.2016.03.003.

[255] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning representations by back-propagating errors". In: *nature* 323.6088 (1986), pp. 533–536.

[256] Amal I. Saba and Ammar H. Elsheikh. "Forecasting the prevalence of COVID-19 outbreak in Egypt using nonlinear autoregressive artificial neural networks". en. In: *Process Safety and Environmental Protection* 141 (Sept. 1, 2020), pp. 1–8. ISSN: 0957-5820. DOI: 10.1016/j.psep.2020.05.029.

[257] Sohail Saif, Priya Das, and Suparna Biswas. "A Hybrid Model based on mBA-ANFIS for COVID-19 Confirmed Cases Prediction and Forecast". en. In: *Journal of The Institution of Engineers (India): Series B* (Jan. 19, 2021). ISSN: 2250-2114. DOI: 10.1007/s40031-021-00538-0.

[258] Sinan Salman and Suzan Alaswad. "Alleviating road network congestion: Traffic pattern optimization using Markov chain traffic assignment". en. In: *Computers & Operations Research* 99 (2018), pp. 191–205. ISSN: 0305-0548. DOI: 10.1016/j.cor.2018.06.015.

[259] Mohd Saqib. "Forecasting COVID-19 outbreak progression using hybrid polynomial-Bayesian ridge regression model". en. In: *Applied Intelligence* (Oct. 23, 2020). ISSN: 1573-7497. DOI: 10.1007/s10489-020-01942-7.

[260] Parsa Sarosh et al. "Artificial Intelligence for COVID-19 Detection – A state-of-the-art review". In: *arXiv:2012.06310 [cs]* (Nov. 25, 2020).

[261] Sayed Tarek and Abdelwahab Walid. "Comparison of Fuzzy and Neural Classifiers for Road Accidents Analysis". In: *Journal of Computing in Civil Engineering* 12.1 (Jan. 1998), pp. 42–47. DOI: 10.1061/(ASCE)0887-3801(1998)12:1(42).

[262] Jonathan W. Schooler. "Metascience could rescue the "replication crisis"". In: *Nature* 515.7525 (2014), 9–9. ISSN: 1476-4687. DOI: 10.1038/515009a.

[263] M. Schuster and K. K. Paliwal. "Bidirectional recurrent neural networks". In: *IEEE Transactions on Signal Processing* 45.11 (Nov. 1997), pp. 2673–2681. ISSN: 1941-0476. DOI: 10.1109/78.650093.

[264] Yann Sellier et al. "Health effects of ambient air pollution: Do different methods for estimating exposure lead to different results?" en. In: *Environment International* 66 (May 2014), pp. 165–173. ISSN: 0160-4120. DOI: 10.1016/j.envint.2014.02.001.

[265] Farah Shahid, Aneela Zameer, and Muhammad Muneeb. "Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM". en. In: *Chaos, Solitons & Fractals* 140 (Nov. 1, 2020), p. 110212. ISSN: 0960-0779. DOI: 10.1016/j.chaos.2020.110212.

[266] Sourabh Shastri et al. "Time series forecasting of Covid-19 using deep learning models: India-USA comparative case study". en. In: *Chaos, Solitons & Fractals* 140 (Nov. 1, 2020), p. 110227. ISSN: 0960-0779. DOI: 10.1016/j.chaos.2020.110227.

[267] Feng Shi et al. "Review of Artificial Intelligence Techniques in Imaging Data Acquisition, Segmentation and Diagnosis for COVID-19". English. In: *IEEE Reviews in Biomedical Engineering* (2020). ISSN: 1937-3333. DOI: 10.1109/RBME.2020.2987975.

[268] Xingjian SHI et al. "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting". In: *Advances in Neural Information Processing Systems 28*. Ed. by C. Cortes et al. Curran Associates, Inc., 2015, pp. 802–810.

[269] Gitanjali R. Shinde et al. "Forecasting Models for Coronavirus Disease (COVID-19): A Survey of the State-of-the-Art". en. In: *SN Computer Science* 1.4 (June 11, 2020), p. 197. ISSN: 2661-8907. DOI: 10.1007/s42979-020-00209-9.

[270] Afshin Shoeibi et al. "Automated Detection and Forecasting of COVID-19 using Deep Learning Techniques: A Review". In: *arXiv:2007.10785 [cs, eess]* (July 27, 2020).

[271] David I Shuman et al. "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains". In: *IEEE Signal Processing Magazine* 30.3 (May 2013), pp. 83–98. ISSN: 1558-0792. DOI: 10.1109/MSP.2012.2235192.

[272] K. Shyam Sunder Reddy, Y. C. A. Padmanabha Reddy, and Ch. Mallikarjuna Rao. "Recurrent neural network based prediction of number of COVID-19 cases in India". en. In: *Materials Today: Proceedings* (Nov. 17, 2020). ISSN: 2214-7853. DOI: `10.1016/j.matpr.2020.11.117`.

[273] Ramon Gomes da Silva et al. "Forecasting Brazilian and American COVID-19 cases based on artificial intelligence coupled with climatic exogenous variables". In: *Chaos, Solitons, and Fractals* 139 (Oct. 2020), p. 110027. ISSN: 0960-0779. DOI: `10.1016/j.chaos.2020.110027`.

[274] Christopher A Sims. "Macroeconomics and reality". In: *Econometrica: journal of the Econometric Society* (1980), pp. 1–48.

[275] D. Singh and C. K. Mohan. "Deep Spatio-Temporal Representation for Detection of Road Accidents Using Stacked Autoencoder". In: *IEEE Transactions on Intelligent Transportation Systems* 20.3 (2019), pp. 879–887. ISSN: 1524-9050. DOI: `10.1109/TITS.2018.2835308`.

[276] Leslie N Smith. "Cyclical learning rates for training neural networks". In: *2017 IEEE winter conference on applications of computer vision (WACV)*. IEEE. 2017, pp. 464–472.

[277] Stanley K. Smith and Terry Sincich. "An Empirical Analysis of the Effect of Length of Forecast Horizon on Population Forecast Errors". In: *Demography* 28.2 (1991), pp. 261–274. ISSN: 0070-3370. DOI: `10.2307/2061279`.

[278] Ewout Steyerberg. *Clinical Prediction Models: A Practical Approach to Development, Validation, and Updating*. en. 2nd ed. Statistics for Biology and Health. Springer International Publishing, 2019. ISBN: 978-3-030-16398-3. DOI: `10.1007/978-3-030-16399-0`.

[279] *Strategies for the surveillance of COVID-19*. en. Apr. 9, 2020.

[280] Abu Sufian et al. "A Survey on Deep Transfer Learning to Edge Computing for Mitigating the COVID-19 Pandemic". In: *Journal of Systems Architecture* 108 (Sept. 2020), p. 101830. ISSN: 1383-7621. DOI: `10.1016/j.sysarc.2020.101830`.

[281] R. Sujath, Jyotir Moy Chatterjee, and Aboul Ella Hassanien. "A machine learning forecasting model for COVID-19 pandemic in India". en. In: *Stochastic Environmental Research and Risk Assessment* 34.7 (July 1, 2020), pp. 959–972. ISSN: 1436-3259. DOI: `10.1007/s00477-020-01827-8`.

[282] Sakhawat Hosain Sumit and Shamim Akhter. "C-means clustering and deep-neuro-fuzzy classification for road weight measurement in traffic management system". en. In: *Soft Computing* 23.12 (2019), pp. 4329–4340. ISSN: 1433-7479. DOI: `10.1007/s00500-018-3086-0`.

[283] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic attribution for deep networks". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 3319–3328.

[284] Yaniv Taigman et al. "DeepFace: Closing the Gap to Human-Level Performance in Face Verification". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. ISSN: 1063-6919. June 2014, pp. 1701–1708. DOI: 10.1109/CVPR.2014.220.

[285] S. K. Tamang, P. D. Singh, and B. Datta. "Forecasting of Covid-19 cases based on prediction using artificial neural network curve fitting technique". In: *Global Journal of Environmental Science and Management* 6.Special Issue (Covid-19) (Aug. 1, 2020), pp. 53–64. ISSN: 2383-3572. DOI: 10.22034/GJESM.2019.06.SI.06.

[286] Leonard J. Tashman. "Out-of-sample tests of forecasting accuracy: an analysis and review". In: *International Journal of Forecasting*. The M3-Competition 16.4 (2000), pp. 437–450. ISSN: 0169-2070. DOI: 10.1016/S0169-2070(00)00065-0.

[287] Mohammad-H. Tayarani N. "Applications of artificial intelligence in battling against covid-19: A literature review". en. In: *Chaos, Solitons & Fractals* 142 (Jan. 1, 2021), p. 110338. ISSN: 0960-0779. DOI: 10.1016/j.chaos.2020.110338.

[288] Shweta Thakur et al. "Prediction for the Second Wave of COVID-19 in India". In: *Big Data Analytics*. Ed. by Ladjel Bellatreche et al. Cham: Springer International Publishing, 2020, pp. 134–150. ISBN: 978-3-030-66665-1.

[289] M. Thatcher and P. Hurley. "A customisable downscaling approach for local-scale meteorological and air pollution forecasting: Performance evaluation for a year of urban meteorological forecasts". en. In: *Environmental Modelling & Software* 25.1 (Jan. 2010), pp. 82–92. ISSN: 1364-8152. DOI: 10.1016/j.envsoft.2009.07.014.

[290] Yuan Tian, Ishika Luthra, and Xi Zhang. "Forecasting COVID-19 cases using Machine Learning models". en. In: *medRxiv* (July 4, 2020). DOI: 10.1101/2020.07.02.20145474.

[291] Anuradha Tomar and Neeraj Gupta. "Prediction for the spread of COVID-19 in India and effectiveness of preventive measures". en. In: *Science of The Total Environment* 728 (Aug. 1, 2020), p. 138762. ISSN: 0048-9697. DOI: 10.1016/j.scitotenv.2020.138762.

[292] O. Torrealba-Rodriguez, R. A. Conde-Gutiérrez, and A. L. Hernández-Javier. "Modeling and prediction of COVID-19 in Mexico applying mathematical and computational models". en. In: *Chaos, Solitons & Fractals* 138 (Sept. 1, 2020), p. 109946. ISSN: 0960-0779. DOI: 10.1016/j.chaos.2020.109946.

[293] Maria Tsikala Vafea et al. "Emerging Technologies for Use in the Study, Diagnosis, and Treatment of Patients with COVID-19". en. In: *Cellular and Molecular Bioengineering* 13.4 (Aug. 1, 2020), pp. 249–257. ISSN: 1865-5033. DOI: 10.1007/s12195-020-00629-w.

[294] Enmei Tu, Nikola Kasabov, and Jie Yang. "Mapping Temporal Variables Into the NeuCube for Improved Pattern Recognition, Predictive Modeling, and Understanding of Stream Data". In: *IEEE Transactions on Neural Networks and Learning Systems* 28.6 (June 2017), pp. 1305–1317. ISSN: 2162-2388. DOI: 10.1109/TNNLS.2016.2536742.

[295] Steffen Uhlig et al. "Modeling projections for COVID-19 pandemic by combining epidemiological, statistical, and neural network approaches". en. In: *medRxiv* (Apr. 22, 2020), p. 2020.04.17.20059535. DOI: 10.1101/2020.04.17.20059535.

[296] PEAN UNION et al. "Directive 2008/50/EC of the European Parliament and of the Council of 21 May 2008 on ambient air quality and cleaner air for Europe". In: *Official Journal of the European Union* (2008).

[297] Arnas Uselis, Mantas Lukoševičius, and Lukas Stasytis. "Localized convolutional neural networks for geospatial wind forecasting". In: *arXiv:2005.05930 [cs, stat]* (May 2020).

[298] T. Vaa, M. Penttinen, and I. Spyropoulou. "Intelligent transport systems and effects on road traffic accidents: state of the art". In: *IET Intelligent Transport Systems* 1.2 (June 2007), pp. 81–88. ISSN: 1751-9578. DOI: 10.1049/iet-its:20060081.

[299] Shashank Reddy Vadyala et al. "Prediction of the Number of COVID-19 Confirmed Cases Based on K-Means-LSTM". In: *arXiv:2006.14752 [physics, q-bio]* (June 25, 2020).

[300] Raju Vaishya et al. "Artificial Intelligence (AI) applications for COVID-19 pandemic". en. In: *Diabetes & Metabolic Syndrome: Clinical ResearchReviews* 14.4 (July 1, 2020), pp. 337–339. ISSN: 1871-4021. DOI: 10.1016/j.dsx.2020.04.012.

[301] Ashish Vaswani et al. "Attention Is All You Need". In: *arXiv:1706.03762 [cs]* (Dec. 5, 2017).

[302] María Vega García and José L. Aznarte. "Shapley additive explanations for NO2 forecasting". en. In: *Ecological Informatics* 56 (Mar. 2020), p. 101039. ISSN: 1574-9541. DOI: 10.1016/j.ecoinf.2019.101039.

[303] Ahmad Waleed Salehi, Preety Baglat, and Gaurav Gupta. "Review on machine and deep learning models for the detection and prediction of Coronavirus". eng. In: *Materials Today. Proceedings* 33 (2020), pp. 3896–3901. ISSN: 2214-7853. DOI: 10.1016/j.matpr.2020.06.245.

[304] Jiawei Wang, Ruixiang Chen, and Zhaocheng He. "Traffic speed prediction for urban transportation network: A path based deep learning approach". In: *Transportation Research Part C: Emerging Technologies* 100 (2019), pp. 372 –385. ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2019.02.002.

[305] Peipei Wang et al. “Time series prediction for the epidemic trends of COVID-19 using the improved LSTM deep learning method: Case studies in Russia, Peru and Iran”. en. In: *Chaos, Solitons & Fractals* 140 (Nov. 1, 2020), p. 110214. ISSN: 0960-0779. DOI: 10.1016/j.chaos.2020.110214.

[306] Senzhang Wang, Jiannong Cao, and Philip S. Yu. “Deep Learning for Spatio-Temporal Data Mining: A Survey”. In: *arXiv:1906.04928 [cs, stat]* (June 2019).

[307] Yingying Wang et al. “The Influence of the Activation Function in a Convolution Neural Network Model of Facial Expression Recognition”. en. In: *Applied Sciences* 10.5 (Jan. 2020), p. 1897. DOI: 10.3390/app10051897.

[308] Yunbo Wang et al. “PredRNN: Recurrent Neural Networks for Predictive Learning using Spatiotemporal LSTMs”. In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 879–888.

[309] P. J. Werbos. “Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences”. PhD thesis. Harvard University, 1974.

[310] WHO. *WHO | Data*. WHO. 2015. URL: http://www.who.int/violence_injury_prevention/road_safety_status/2015/GSRRS2015_data/en/.

[311] Michał Wieczorek, Jakub Siłka, and Marcin Woźniak. “Neural network powered COVID-19 spread forecasting model”. en. In: *Chaos, Solitons & Fractals* 140 (Nov. 1, 2020), p. 110203. ISSN: 0960-0779. DOI: 10.1016/j.chaos.2020.110203.

[312] Michał Wieczorek et al. “Real-time neural network based predictor for cov19 virus spread”. en. In: *PLOS ONE* 15.12 (2020), e0243189. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0243189.

[313] Christopher K. Wikle, Andrew Zammit-Mangion, and Noel Cressie. *Spatio-Temporal Statistics with R*. 1st ed. Boca Raton, Florida : CRC Press, [2019]: Chapman and Hall/CRC, 2019. ISBN: 978-1-351-76972-3. DOI: 10.1201/9781351769723. URL: https://www.taylorfrancis.com/books/9780429649783.

[314] Christopher K. Wikle, Andrew Zammit-Mangion, and Noel Cressie. *Spatio-Temporal Statistics with R*. en. 1st ed. Boca Raton, Florida : CRC Press, [2019]: Chapman and Hall/CRC, Feb. 2019. ISBN: 978-1-351-76972-3. DOI: 10.1201/9781351769723. URL: https://www.taylorfrancis.com/books/9780429649783.

[315] Cort J. Willmott and Kenji Matsuura. “Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance”. In: *Climate Research* 30.1 (2005), pp. 79–82. ISSN: 0936-577X.

[316] Ian H. Witten et al. "Chapter 10 - Deep learning". en. In: *Data Mining (Fourth Edition)*. Morgan Kaufmann, Jan. 1, 2017, pp. 417–466. ISBN: 978-0-12-804291-5. DOI: 10.1016/B978-0-12-804291-5.00010-6.

[317] Sean L. Wu et al. "Substantial underestimation of SARS-CoV-2 infection in the United States". en. In: *Nature Communications* 11.1 (Sept. 9, 2020), p. 4507. ISSN: 2041-1723. DOI: 10.1038/s41467-020-18272-4.

[318] Tianshu Wu et al. "A Multiple SVR Approach with Time Lags for Traffic Flow Prediction". In: ISSN: 2153-0009, 2153-0017. Oct. 2008. DOI: 10.1109/ITSC.2008.4732663.

[319] Yuankai Wu et al. "A hybrid deep learning based traffic flow prediction method and its understanding". en. In: *Transportation Research Part C: Emerging Technologies* 90 (May 2018), pp. 166–180. ISSN: 0968-090X. DOI: 10.1016/j.trc.2018.03.001.

[320] Zonghan Wu et al. "Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks". In: *KDD 2020* (May 2020).

[321] Laure Wynants et al. "Prediction models for diagnosis and prognosis of covid-19: systematic review and critical appraisal". In: *BMJ* 369 (2020). DOI: 10.1136/bmj.m1328.

[322] Pengpeng Xu and Helai Huang. "Modeling crash spatial heterogeneity: Random parameter versus geographically weighting". In: *Accident Analysis & Prevention* 75 (2015), pp. 16–25. ISSN: 0001-4575. DOI: 10.1016/j.aap.2014.10.020.

[323] Nesrine Ben Yahia, Mohamed Dhiaeddine Kandara, and Narjes Bellamine Ben Saoud. *Deep Ensemble Learning Method to Forecast COVID-19 Outbreak*. Tech. rep. In Review, May 21, 2020. DOI: 10.21203/rs.3.rs-27216/v1.

[324] Bingjie Yan et al. "An Improved Method for the Fitting and Prediction of the Number of COVID-19 Confirmed Cases Based on LSTM". en. In: *Computers, MaterialsContinua* 64.3 (June 30, 2020), pp. 1473–1490. ISSN: 1546-2226. DOI: 10.32604/cmc.2020.011317.

[325] Hao-Fan Yang, Tharam S. Dillon, and Yi-Ping Phoebe Chen. "Optimized Structure of the Traffic Flow Forecasting Model With a Deep Learning Approach". In: *IEEE Transactions on Neural Networks and Learning Systems* 28.10 (Oct. 2017), pp. 2371–2381. ISSN: 2162-2388. DOI: 10.1109/TNNLS.2016.2574840.

[326] Kui Yang, Xuesong Wang, and Rongjie Yu. "A Bayesian dynamic updating approach for urban expressway real-time crash risk evaluation". In: *Transportation Research Part C: Emerging Technologies* 96 (2018), pp. 192–207. ISSN: 0968-090X. DOI: 10.1016/j.trc.2018.09.020.

[327] Zifeng Yang et al. "Modified SEIR and AI prediction of the epidemics trend of COVID-19 in China under public health interventions". eng. In: *Journal of Thoracic Disease* 12.3 (Mar. 2020), pp. 165–174. ISSN: 2072-1439. DOI: 10.21037/jtd.2020.02.64.

[328] Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. “On early stopping in gradient descent learning”. In: *Constructive Approximation* 26.2 (2007), pp. 289–315.

[329] Quanzeng You et al. “Image Captioning With Semantic Attention”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[330] Byeonghyeop Yu, Yongjin Lee, and Keemin Sohn. “Forecasting road traffic speeds by considering area-wide spatio-temporal dependencies based on a graph convolutional neural network (GCN)”. en. In: *Transportation Research Part C: Emerging Technologies* 114 (May 2020), pp. 189–204. ISSN: 0968-090X. DOI: `10.1016/j.trc.2020.02.013`.

[331] Y. Yu, M. Xu, and J. Gu. “Vision-based traffic accident detection using sparse spatio-temporal features and weighted extreme learning machine”. In: *IET Intelligent Transport Systems* 13.9 (2019), pp. 1417–1428. ISSN: 1751-956X. DOI: `10.1049/iet-its.2018.5409`.

[332] Zhuoning Yuan, Xun Zhou, and Tianbao Yang. “Hetero-ConvLSTM: A Deep Learning Approach to Traffic Accident Prediction on Heterogeneous Spatio-Temporal Data”. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD ’18*. the 24th ACM SIGKDD International Conference. London, United Kingdom: ACM Press, 2018, pp. 984–992. ISBN: 978-1-4503-5552-0. DOI: `10.1145/3219819.3219922`.

[333] Novanto Yudistira. “COVID-19 growth prediction using multivariate long short term memory”. en. In: *Preprint* (May 10, 2020).

[334] Seid Miad Zandavi, Taha Hossein Rashidi, and Fatemeh Vafaee. “Forecasting the Spread of Covid-19 Under Control Scenarios Using LSTM and Dynamic Behavioral Models”. In: *arXiv:2005.12270 [physics]* (May 24, 2020).

[335] Abdelhafid Zeroual et al. “Deep learning methods for forecasting COVID-19 time-Series data: A Comparative study”. en. In: *Chaos, Solitons & Fractals* 140 (Nov. 1, 2020), p. 110121. ISSN: 0960-0779. DOI: `10.1016/j.chaos.2020.110121`.

[336] Guangnan Zhang, Kelvin K. W. Yau, and Guanghan Chen. “Risk factors associated with traffic violations and accident severity in China”. In: *Accident Analysis & Prevention* 59 (2013), pp. 18–25. ISSN: 0001-4575. DOI: `10.1016/j.aap.2013.05.004`.

[337] Yang Zhang and Tao Cheng. “Graph deep learning model for network-based predictive hotspot mapping of sparse spatio-temporal events”. en. In: *Computers, Environment and Urban Systems* 79 (Jan. 2020), p. 101403. ISSN: 0198-9715. DOI: `10.1016/j.compenvurbsys.2019.101403`.

[338] Yang Zhang et al. “A novel residual graph convolution deep learning model for short-term network-based traffic forecasting”. In: *International Journal of Geographical Information Science* 34.5 (May 2020), pp. 969–995. ISSN: 1365-8816. DOI: `10.1080/13658816.2019.1697879`.

[339] Zhenhua Zhang et al. "A deep learning approach for detecting traffic accidents from social media data". In: *Transportation Research Part C: Emerging Technologies* 86 (2018), pp. 580–596. ISSN: 0968-090X. DOI: `10.1016/j.trc.2017.11.027`.

[340] Bendong Zhao et al. "Convolutional neural networks for time series classification". In: *Journal of Systems Engineering and Electronics* 28.1 (Feb. 2017), pp. 162–169. ISSN: 1004-4132. DOI: `10.21629/JSEE.2017.01.18`.

[341] Zhuowen Zhao, Kieran Nehil-Puleo, and Yangzhi Zhao. "How well can we forecast the COVID-19 pandemic with curve fitting and recurrent neural networks?" In: *Preprint* (May 18, 2020). DOI: `10.1101/2020.05.14.20102541`.

[342] M. Zheng et al. "Traffic Accident's Severity Prediction: A Deep-Learning Approach-Based CNN Network". In: *IEEE Access* 7 (2019), pp. 39897–39910. ISSN: 2169-3536. DOI: `10.1109/ACCESS.2019.2903319`.

[343] Nanning Zheng et al. "Predicting COVID-19 in China Using Hybrid AI Model". In: *IEEE Transactions on Cybernetics* (2020). DOI: `10.1109/TCYB.2020.2990162`.

[344] Zhenpeng Zhou and Xiaocheng Li. "Graph Convolution: A High-Order and Adaptive Approach". In: *arXiv:1706.09916 [cs, stat]* (Oct. 2017).

[345] Qiaomu Zhu et al. "Wind Speed Prediction with Spatio–Temporal Correlation: A Deep Learning Approach". en. In: *Energies* 11.4 (Apr. 2018), p. 705. DOI: `10.3390/en11040705`.