Using Deep Transformer Based Models to Predict Ozone Levels

Manuel Méndez, Carlos Montero, and Manuel Núñez(✉)

Universidad Complutense de Madrid, Madrid, Spain
{manumend,cmonte09,manuelnu}@ucm.es

Abstract. Ozone ($O3$) is an air pollutant that has harmful effects in human health when its concentration exceeds a certain level. Therefore, it is important to advance in methods that can appropriately predict $O3$ levels. In this paper we present a new model to estimate 4 h, 12 h, 24 h, 48 h and 72 h ahead $O3$ concentration levels. We rely on Deep Transformer Networks. Interestingly enough, these models were originally developed to be used in Natural Language Processing applications but we show that they can be successfully used in classification problems. In order to evaluate the usefulness of our model, we applied it to predict $O3$ levels in the centre of Madrid. We compare the results of our model with four baseline models: two LSTMs and two MLPs. Accuracy (Acc) and Balanced Accuracy (BAC) are the metrics employed to evaluate the goodness of all the models. The results clearly show that our Deep Transformer based Network obtains the best results.

Keywords: Air quality prediction · Deep learning · Transformer networks

1 Introduction

Air pollution is one of the major problems currently faced by humanity. It causes seven million deaths every year according to international organisms [5]. Pollutants like ozone ($O3$), nitrogen dioxide ($NO2$), sulphur dioxide ($SO2$), carbon monoxide (CO) and particulates matter ($PM2.5$, $PM10$) are some of the most common air pollutants [4]. They are also the pollutant included in the air quality index measurement. In this paper we focus on $O3$ (ozone). Ozone is a colourless gas located in the atmosphere. It is one of the most common existing pollutants and, as such, it is one of the pollutants taken into account to determine the air quality index. The exposure to this pollutant has severe effects in human health such as eyes and nose irritation and inflammation, lung function reduction, exacerbation of respiratory diseases and increased susceptibility to diseases infection,

This work has been supported by the Spanish MINECO/FEDER projects FAME (RTI2018-093608-B-C31) and AwESOMe (PID2021-122215NB-C31) and the Region of Madrid project FORTE-CM (S2018/TCS-4314), co-funded by EIE Funds of the European Union.

N. T. Nguyen et al. (Eds.): ACIIDS 2022, LNAI 13757, pp. 169–182, 2022.
https://doi.org/10.1007/978-3-031-21743-2_14

among others [4]. Ozone concentration levels depend on complex processes that happen in the atmosphere. Precursor gases, such as nitrogen oxides, are chemically transformed into ozone when they are exposed to solar light. Since sunlight is needed for ozone formation, its concentration highly depends on the time of the day and on the current meteorological conditions. Moreover, human emission of these precursor pollutants by fabrics and traffic have affected the increase of ozone concentration in the last decades. Most major cities around the world have specific protocols to deal with high concentrations of pollutants. In the case of Madrid, used in this paper as case study, when the $O3$ concentration exceeds a certain level, the authorities take measures to reduce the $O3$ effects in population. These measures range from recommendations to reduce physical exercise outdoors for vulnerable people to prohibitions of outdoor activities, specially sport activities [2]. Therefore, it is very important to be able to estimate future $O3$ levels in order to alert the population of possible future recommendations or restrictions.

We present a new model to predict the $O3$ concentration level. Our approach applies a novel Transformers-based model to predict the $O3$ concentration level. Unlike classical time series forecasting models, Transformers do not process the data ordered. On the contrary, they process the entire sequence and use self-attention techniques to find dependencies between variables. We considered an optimised Transformers-based model and our preliminary experiments revealed that it was a good candidate to overcome the accuracy and the balanced accuracy of usual time series classification networks such as MLP or LSTM.

In order to evaluate the usefulness of our proposal, we compare its results with the ones produced by four neural network baselines models. We also make an analysis of the variation of the model accuracy and balanced accuracy depending on the modification of three hyperparameters in short-term and in long-term cases. For this, we use as case of study the task of predicting $O3$ levels in the centre of Madrid for the next 4, 12, 24, 48 and 72 h. We consider a total of fourteen predictors variables. Our results show that our proposal is better than the baseline models based on the evaluation metrics. In average, the balanced accuracy of our proposal is 4.3% better than the one corresponding to the best baseline model.

The rest of the paper is organised as follows. Section 2 reviews related work. In Sect. 3 we present background concepts such as the baseline models and different evaluation metrics that we use. In Sect. 4 we present the main characteristics of the problem that we want to solve and of the model that we construct to confront the problem. In Sect. 5 we present our experiments and discuss the obtained results. Finally, in Sect. 6, we give conclusions and outline some directions for future work.

2 Related Work

Several studies have used either statistical machine learning techniques or deep learning models to forecast pollutant concentrations. Paoli *et al.* [19] develop an optimised MLP network to forecast $O3$ concentration in Corsica. This model is more accurate than other baseline models and properly detects ozone peaks. Li *et al.* [15] propose a Random Forest model, leveraging its capacity to work with numerical and non-numerical variables, to predict the concentration of three air pollutants. Other models have been developed to forecast pollutant concentrations such as an improved ARIMA (Liu *et al.*) [16], LSTM networks (Seng *et al.* [21]) and multi-linear regression (Jato-Espino *et al.*) [13]. In addition to machine learning techniques, collective information can be compiled to forecast air quality. Palak *et al.* [18] present a collective framework to predict air pollution in places where no meters are available. The combination of monitoring and CEP is also a good approach to forecast air quality. In this line, Díaz *et al.* [8] considered Petri Nets while Corral-Plaza *et al.* [7] and Semlali *et al.* [20] used an IoT approach.

Our approach is based on Transformer Networks. Transformers were proposed by Vaswani *et al.* [22]. Originally, they were developed as a Natural Language Processing tool that improves classical LSTM networks and Recurrent Neural Networks (RNN) in tasks such as text classification and translation. Its potential is based on self-attention layers, which estimate the attention weights between input variables. In recent years, Transformers have been used to solve tasks in other fields such as image recognition, multi-class classification and time series prediction. Wu *et al.* [23] present a deep model based on Transformers to influenza-like illness forecasting. Results show that the proposed method is more precise than baseline methods such as LSTM or ARIMA. Dosovitsky *et al.* [9] show that a pure Transformer application can overcome classical CNNs in image classification tasks.

To the best of our knowledge, Deep Transformer based Models have not been used to analyse air pollutants. Although machine learning techniques have been used to forecast ozone concentration values [6,19] (that is, as part of a regression model), we are not aware of their use to predict ozone levels as such (that is, as part of a classification model).

3 Preliminaries

In this section we will review some concepts that we will use along the paper. Specifically, we will discuss the baseline models that will be used to compare our approach with and the evaluation metrics that will be used to measure different models quality.

3.1 Baseline Models

In order to assess the usefulness of our approach, we will compare it with two classical algorithms that are very suitable to solve the same problem: *LSTM networks* and *MLP networks*. Although these models were defined some time ago, they are currently and frequently used both to predict the behaviour of complex systems and as baseline models [10,14,17].

MLP Networks [12]. An MLP network is a computational model inspired by a human brain whose objective is to find relationships between data. It is composed by three types of layers: input layer, hidden layer and output layer. The input layer receives the input data to be processed. A number of hidden layers are placed between the input and output layers. Data flows from input to output in forward direction. Each layer is composed by a number of simple processing elements called *neurons*. Neurons are trained using the back-propagation learning algorithm to minimise a loss function.

The mathematical operations that occur in every neuron in hidden and output layers are, respectively, given by the following expressions:

$$H_x = f(b_1 + W_1 \cdot x) \qquad O_x = f'(b_2 + H_x \cdot W_2)$$

being x an input vector, b_1 and b_2 bias vectors, W_1 and W_2 weight matrices and f and f' activation functions. Usual activation functions are the RELU and sigmoid functions.

$$RELU(a) = \max(0, a) \qquad Sig(a) = \frac{1}{1 + e^{-a}},$$

where a is the input data.

Hyperparameters such as the *number of hidden layers* (h), *number of neurons in each hidden layer* (h_n), *number of epochs* (ep), *dropout rate* (dr), *learning rate* (lr), and batch size (bs) are optimised to get the best accuracy in MLP networks.

LSTM Networks [24]. An RNN network does not have a defined layers structure. Actually, it allows random connections between neurons, developing temporality and providing memory to the network. This makes RNNs well suited algorithms in fields such as Natural Language Processing and sequence data processing. However, if a long context is needed, we have the long-term dependencies problem, that is, the gradual forgetfulness of previous information of the network. In order to solve it, LSTM networks were developed.

The key to LSTMs is the cell state (C_t) that runs straight down the entire chain. The information flows along it almost without modifications. The LSTM either updates or discards information in the cell state by using structures called *gates*. These gates are composed by a sigmoid layer, which outputs a number between 0 and 1 that indicates how much information must be let through, and a multiplicative element. LSTM cells have three gates: input gate (i_t), forget gate (f_t) and output gate (o_t). The mathematical operations that occur in each gate are as follows:

$$\begin{gathered}
f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
\tilde{C_t} = tanh(W_c \cdot [h_{t-1}, x_t] + b_C) \\
C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C_t} \\
o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \\
h_t = o_t \cdot tanh(C_t)
\end{gathered}$$

where x_t is the input data in time t, h_t is the hidden state in time t, each W_x is a weight matrix and each b_x is a bias vector.

Hyperparameters such as the *number of neurons in LSTM layer* (n), *number of epochs* (ep), *dropout rate* (dr), *learning rate* (lr) and batch size (bs) are optimised to get the best accuracy in the LSTM network.

3.2 Performance Evaluation Metrics

We will consider accuracy (Acc) because it is the most used evaluation measure in multi label classification models [11]. In addition, as in all cases our data is imbalanced, we will also use Balanced Accuracy (BAC), which gives the same weight to all the categories. Given $m \in \mathbb{N}$, let M be an $m \times m$ confusion matrix. The formal definitions of these measures are:

$$Acc = \frac{\sum\limits_{k=1}^{m} M_{k,k}}{\sum\limits_{k=1}^{m}\sum\limits_{i=1}^{m} M_{k,i}} BAC = \frac{\sum\limits_{k=1}^{m} \frac{M_{k,k}}{\sum_{i=1}^{m} M_{k,i}}}{m}$$

Intuitively, Acc is the ratio between the number of correctly predicted observations and total number of observations, while BAC computes these ratios for each category.

4 Problem Description and Our Model

In this section we present the main concepts underlying our case of study. We also present a definition and description of our Deep Transformer based Network. Finally, we set the hyperparameters of each baseline model that will be used to compare with our model.

4.1 Problem Description

In order to evaluate the proposal model, we decided to use pollutant data because its concentration usually depends on a great amount of factors such as meteorology, industrial emissions, transportation and use of chemical products. Among the different pollutants, we choose ozone because it has a behaviour along the year that diverges from the rest of pollutants influencing the Air Quality Index. We wish to evaluate the goodness of our proposal in a classification problem and using a previous *standard* that we cannot influence. Therefore, we categorise ozone values but take into account its hazard as determined by official standards of Madrid Council [2].

The ozone concentration level predicting problem is formulated as a supervised machine learning task. Specifically, our goal is to predict the category of the target variable y in k hours time, that is, $y(h + k)$. We will use a vector of 56 predictor variables, 14 by hour, and using the last 4 available values. We have 12 continuous hourly predictor variables and 2 (both calendar variables) categorical ones. The target variable is a categorical one. Table 1 shows all the predictor variables (* denotes a categorical variable).

Table 1. Predictor variables by type

Type	Variables
Pollutant	$NO, CO, NO2, NOX, SO2, O3$
Meteorological	$rain, maxtemp, mintemp, medtemp, maxpress, minpress$
Calendar	$week\ day^*, type\ of\ day^*$

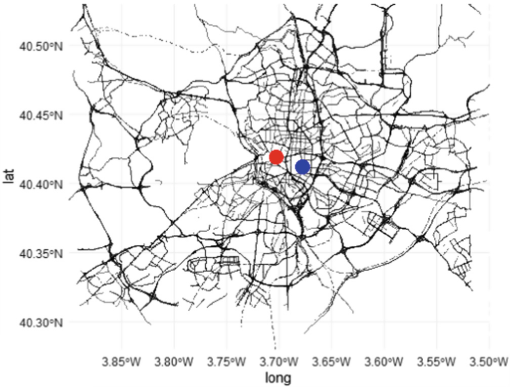


Fig. 1. City of Madrid. The red mark points at 'Plaza del Carmen Station' while the blue one points at 'El Retiro Station'. (Color figure online)

Our proposal, as well as the baseline models, will be evaluated in a real case study. Specifically, we consider data collected in Madrid from 01/01/2017 to 31/07/2021. Pollutant data is hourly collected by dedicated sensors, placed in stations distributed in the city, and it is publicly available at the Madrid City Hall website [2] database. We also use this database to extract calendar data. In addition, meteorological data can be extracted from AEMET [1], the State Meteorology Agency of Spain. We transform daily variables in hourly variables to standardise all of them. We consider two stations located in the centre of Madrid (see Fig. 1): 'Plaza del Carmen Station' and 'El Retiro'.

Data has been pre-processed by inputting previous observation variable values in not available and outliers values. All the variables have also been scaled. Ozone continuous data runs from 0 μg/m^3 to 373 μg/m^3. Following the guideline of Madrid City Hall, we classify it in a three class target variable. In Table 2 we show the categories chosen and the number of existing observations in each of them. We obtain a final data-set with 38.064 observations.

Table 2. Ozone levels

Category (values)	Ozone state	Number of observations
0 (0–60)	Good	26.985
1 (60–120)	Medium	10.509
2 (>120)	Harmful	570

4.2 Deep Transformer Based Models

In this section we describe the main characteristics of our model, which is based on Deep Transformer Networks [22]. The model architecture is composed by a number of Transformer encoder networks (*num_transformers*) and a final MLP network to make the classification. Each transformer encoder is composed by a *normalisation* layer, which applies a transformation to maintain the mean of the previous layer activation close to 0 and the standard deviation close to 1. Then, we add the essence of the Transformer model: an *attention* layer. It uses an attention mechanism to learn the contextual relation between variables. This mechanism avoids the requirement of the recurrent connections in the neural network. This layer has a set of hyperparameters, which must be controlled, such as the number of attention heads (*num_head*). A number of heads greater than 1 (Multi-Head Attention) allows the model to jointly attend to information from different sub-spaces of representation at different positions [22]. The size of each one (*head_size*) and the dropout probability (*dropout*) are other hyperparameters to consider. The next step is to add the feed forward part of the transformer encoder, which is composed by a *normalisation layer* and by two one dimension *convolutional layers* with a kernel size equal to one. A dropout probability is applied in the first convolutional layer, which works with a ReLU activation function. The convolutional layers create a convolutional kernel with the normalisation layer, in this case, in a temporal dimension $t = 1$. In the first convolutional layer, the number of output filters is chosen by the *filters* hyperparameter. In the second one, the number of output filters is equal to the dimension of the input shape. In order to reduce the output tensor of the set of Transformer Encoders, we add a one dimension *Average Pooling Layer*. Finally, we add an MLP network to make the classification. This MLP uses ReLU as activation function in the hidden layer and softmax as activation function in the output layer. We also apply here a dropout probability (*mlp_dropout*). MLP has just one hidden layer. The number of neurons in the hidden layer of the MLP (*hid_layer*) is a hyperparameter to consider.

In order to train the network we use the gradient descent learning method. We apply also an *Early Stopping*, which is a regulation method to stop the training when the error diminution between two consecutive iterations is less than a previously set threshold (usually, 0.0001).

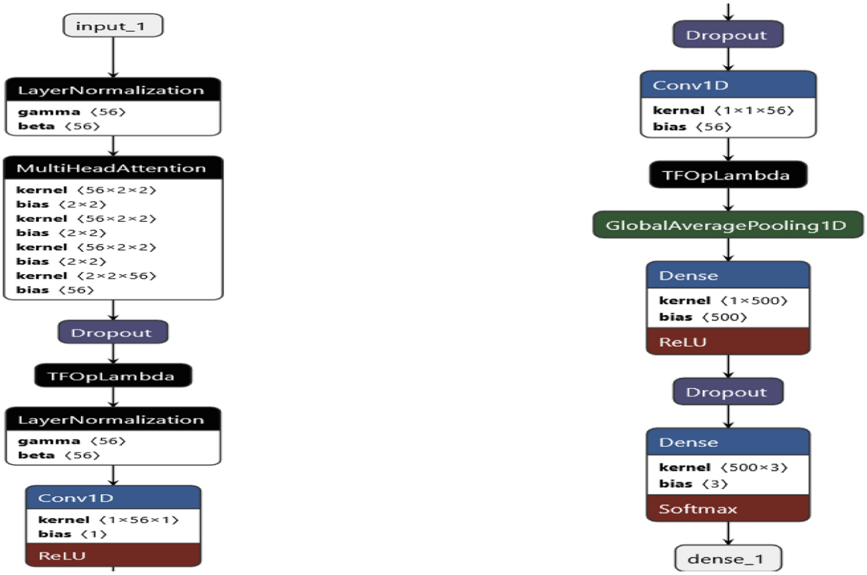
The chosen loss function has been the *sparse categorical cross-entropy* and we consider *Adam* as weights optimiser. The number of epochs and the batch size will be two of the hyperparameters to fit.

We use python, specifically the *keras* library, to implement our model.

In Fig. 2 we use *Netron* [3] to show the structure of our Deep Transformer based Network with the following hyperparameters: $num\_transformers = 1$, $num\_head = 2$, $head\_size = 2$, $dropout = 0.2$, $filters = 1$, $mlp\_dropout = 0.15$ and $hid\_layer = 500$. In order to reduce the size of the graphical representation, we use low hyperparameters values. The interested reader can visit https://github.com/MMH1997/TransformerNetworks where it is possible to see the architecture with other hyperparameters and deeply analyse each component of the model.

4.3 MLP and LSTM Networks

In this section we describe the baseline models with which we will compare our proposal. In order to make an unbiased comparison, we compare our proposal with two LSTMs and two MLPs, with respectively more/less trainable parameters than our proposal. We use the *keras* library in python to implement these models.



(a) Part a.

(b) Part b.

Fig. 2. Deep transformer based model structure

- *MLP networks.* The first MLP model (*MLP1*) has $h = 5$ hidden layers. The number of neurons in each hidden layer are 512, 256, 128, 32 and 32. It results in a total of 198.691 trainable parameters. The second MLP model (*MLP2*) also has $h = 5$ hidden layers. The number of neurons in each hidden layer are 64, 64, 16, 8 and 8. It results in a total of 9.083 trainable parameters.
  Both models have the following hyperparameters: $ep = 25$, $dr = 0.2$, $lr = 0.0001$ and $bs = 16$. Each hidden layer is activated by a ReLU function and the output layer by a softmax function. Both output layers have three neurons (each of them returns the probability of each category). In both models, the loss function has been *sparse categorical cross-entropy*.

- *LSTM networks.* Our first LSTM model (*LSTM1*) has $n = 150$ neurons in the LSTM layer. This layer is connected to three hidden layers having, respectively 15, 5 and 5 neurons. Hidden layers are activated by a ReLU function. Again, the output layer has three neurons (each of them returns the probability of each category) and is activated by a softmax function. This model has a total of 123.593 trainable parameters. Our second LSTM model (*LSTM2*) has $n = 12$ neurons in the LSTM layer. This layer is connected to an output layer that has three neurons and is activated by a softmax function. This model only has 3.251 trainable parameters.
  Both models have the following hyperparameters: $ep = 25$, $dr = 0.2$, $lr = 0.0001$ and $bs = 16$. In these two models, the loss function has also been the *sparse categorical cross-entropy*.

Table 3. *Acc* by model and time in advance

Model	4 h	12 h	24 h	48 h	72 h	Average
Proposal	**0.891**	**0.730**	**0.772**	0.753	**0.753**	**0.776**
$LSTM1$	0.872	0.642	0.753	0.716	0.700	0.729
$LSTM2$	0.881	0.651	0.762	0.741	0.722	0.744
$MLP1$	**0.891**	0.707	0.768	0.747	0.736	0.765
$MLP2$	0.882	0.701	0.771	**0.757**	0.730	0.764

Table 4. *BAC* by model and time in advance

Model	4 h	12 h	24 h	48 h	72 h	Average
Proposal	0.816	**0.471**	**0.570**	**0.484**	**0.480**	**0.540**
$LSTM1$	0.730	0.346	0.467	0.428	0.411	0.448
$LSTM2$	0.705	0.351	0.491	0.467	0.446	0.467
$MLP1$	**0.819**	0.462	0.520	0.471	0.451	0.518
$MLP2$	0.686	0.429	0.517	0.482	0.437	0.496

5 Experiments

In this section we present the experiments that we performed to validate our model. In all the experiments we used 75% of the observations as training set and 25% of the observations as testing set. Since the difference between the results is not very large, in order to reduce the error we compute the average of 20 repetitions of the experiment. The programming code and data used in these experiments are freely available at https://github.com/MMH1997/TransformerNetworks.

5.1 Comparison Between Models

We apply the proposed model and the baseline models to predict the categories in the next 4, 12, 24, 48 and 72 h. In this experiment, we use the following hyperparameters in the Deep Transformer based Model: $num\_transformers = 2$, $num\_head = 15$, $head\_size = 5$, $dropout = 0.2$, $filters = 25$, $mlp\_dropout = 0.15$, $hid\_layer = 5000$, $bs = 8$ and $epochs = 25$. This choice results in a total of 65.375 trainable parameters.

Comparing the model with the previously defined MLP and LSTM networks, we appreciate that, in general, our developed model obtains better results, particularly in long-term cases. In terms of accuracy (see Table 3), our model obtains higher accuracy than all the baselines models, except in the 4 h case, where it obtains the same accuracy as $MLP1$, and in the 12 h case, where our model is slightly worse than $MLP2$. In terms of Balanced Accuracy (see Table 4), the proposed model obtains the best accuracy in all cases except in the 4 h case, where our model is worse than $MLP1$. It is worth to mention the 24 h case, where the difference between the BAC of the proposed model and the one of the baseline models is greater than 5 points. We think that this is due, in part, to the capability of our model to detect minority class cases, while the baseline models are not able to detect them when predicting 24 h in advance.

By the resolve of the task, the presence of a minority class in our data which implies the called *imbalanced class problem* must taken into account. We have tried to solve this problem by applying typical techniques to diminish it such as over-sampling, under-sampling or re-sampling. However, none of them were effective, probably due to the high number of variables and the dynamism in data. This ineffectiveness was reflected on the inability of all models (proposed and baseline) to get a BAC higher than 0.33, that is, models just classify all the observations in the same category.

The obtained values show the typical increase of Acc and BAC as time in advance is reduced, that is, we expect better results in predictions 4 h in advance than in predictions in a longer time. However, this pattern does not hold if we consider the 12 h case. In fact, there is a clear reduction of Acc and BAC in all models with respect to the 24 h or 48 h cases. After a careful review of the factors that influence ozone concentration, we realised that meteorological data, in particular and as we advanced in the introduction of the paper the presence/absence of solar light, are very relevant. In more technical terms, the correlation between the values of the predictors at time t and the ones at times $t - 24$ and $t - 48$ is higher than the corresponding to time $t - 12$. In fact, the difference of several predictor variables in the 12 h case is usually high, not only concerning meteorological variables but also other pollutant concentrations (NO, CO, $SO2$, $NO2$, NOX). For example, in most cases, the average value of these pollutants at noon is the double than the average value at midnight.

In the next experiments, we will analyse the optimisation of the proposal model hyperparameters.

5.2 Hyperparameters Optimisation

In this section we will analyse the evolution of BAC and Acc depending of the values of three hyperparameters: *num_head*, *head_size* and $filters$. We performed two sets of experiments: prediction 4 h in advance and prediction 24 h in advance. We chose these two cases because they are the shortest term and the most *typical* prediction (what will happen the next day at the same time). It is worth to mention that our preliminary experiments showed a similar behaviour for, on the one hand, all short term cases and, on the other hand, all long term cases.

The following values were chosen to evaluate each hyperparameter:

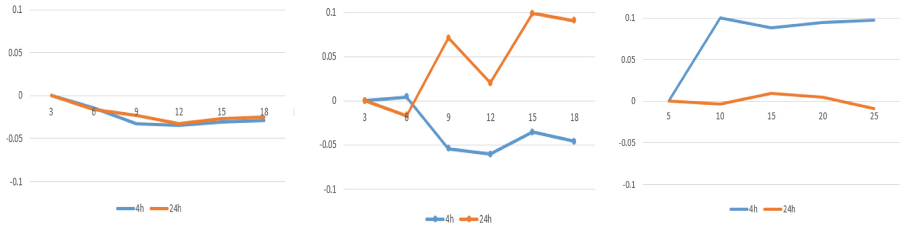
- *num_head*: 3, 6, 9, 12, 15, 18;
- *head_size*: 3, 6, 9, 12, 15, 18;
- $filters$: 5, 10, 15, 20, 25.

The rest of hyperparameters used in this experiment are set to: $num\_transformers = 1$, $dropout = 0.2$, $mlp\_dropout = 0.15$, $hid\_layer = 5000$, $bs = 8$ and $epochs = 25$. Finally, the number of trainable parameters in the combinations range from 27.947 to 101.712.

We have evaluated BAC and Acc in all the possible combinations of the hyperparameters values previously mentioned. In order to provide a unique result for each experiment, we compute the average value of the metrics evaluated for each hyperparameter value. For example, if we fix $head\_size = 6$, we compute the average of all the values returned from the 30 experiments corresponding to all the combinations of *num_head* and $filters$ such that $head\_size = 6$. In the two cases studies, Acc values remain almost constant with the hyperparameters modifications. Therefore, we focus our analysis on the variations of BAC values.

The experiments corresponding to the 4 h in advance case clearly show that low values of *head_size* and *num_head* return higher values of BAC. If we consider $filters$, we observe that the maximum BAC values are achieved when $filters = 10$. In higher $filters$ values, BAC values remain constant. Interestingly enough, the absolute maximum BAC value (0.8228) is achieved in a combination of hyperparameters values that does not correspond to the general conclusions: $head\_size = 18$, $num\_head = 15$ and $filters = 25$.

If we consider the 24 h in advance case, we observe that an increase of *head_size* values suggest a small decrease of BAC. Unlike the previous case, maximum BAC values are achieved when $num\_head = 15$. For higher values, BAC seems remain constant. Maximum BAC values are achieved when $filters = 15$. However, experiments show that changes in $filters$ values do not imply significant modifications in BAC. Unlike the previous case, the absolute maximum BAC value seems correspond to the general conclusions. It is achieved when $head\_size = 9$, $num\_head = 15$ and $filters = 5$.



(a) *head_size* (b) *num_head* (c) *filters*

Fig. 3. *BAC* percentage change by hyperparameter value in short-term and long-term cases.

In Fig. 3 we show how the obtained values vary. From these results we can extract the following conclusions:

- The variation of *BAC* values depending on the performance of *head_size* is similar in both the short term and long term cases.
- In short term cases, an increase of *num_head* suggests an increase of *BAC*. However, in long term cases, the increase in the value of the hyperparameter suggests a decrease of *BAC* values.
- In long term cases, the modification of *filters* values seems to have no effect in *BAC*. In contrast, in short term cases, its increase implies an increase of *BAC*.

6 Conclusions and Future Work

In this paper, we have proposed a novel Deep Transformer based Network to classify the ozone levels in the centre of Madrid using fourteen predictor variables. Our experiments show that this model overcomes, in general terms, the accuracy and the balanced accuracy of four baselines neural network models. The capability of the proposed model to detect minority class observations, particularly, in 24 h in advance case is one of the main advantages of proposed model respect to the baseline methods.

We have also made an analysis of three hyperparameters of the proposal model in short term and in long term cases. This analysis suggests us that the behaviour of balanced accuracy depending on hyperparameters modifications differs between long term and short term cases.

We consider some lines for future work. First, in order to increase accuracy, we would like to perform a deeper analysis on the optimisation of the hyperparameters, in particular, concerning whether we need similar adjustments for short-term and long-term prediction. For example, in Fig. 3 we can see that $head_size$ has a similar behaviour in both cases, while it is very different for *num_head* and *filters*. Second, taking into account the high quality of the proposed model, we would like to produce a similar model for other data. In particular, we would like to apply this model to the rest of pollutants included in the air quality index measurement so that it is possible to classify them by levels. Combining this idea with Complex Event Processing technologies, air quality index could be accurately predicted. We would also like to adapt our model to other air quality indexes that, in particular, might have data in a format that it is not compatible with the one that we have used. Finally, we would like to compare our proposal with other classification models such as Random Forest, ARIMA and Support Vector Machine.

References

1. AEMET Open Data. https://opendata.aemet.es/centrodedescargas/productosAEMET?. Accessed 25 Oct 2021
2. City of Madrid. https://www.mambiente.madrid.es/opencms/export/sites/default/calaire/Anexos/Procedimiento_ozono.pdf. Accessed 20 Oct 2021
3. Netron open source tool. https://netron.app/. Accessed 15 Nov 2021
4. New South Wales Government. https://www.health.nsw.gov.au/environment/air/Pages/common-air-pollutants.aspx. Accessed 20 Oct 2021
5. World Health Organisation. https://www.who.int/health-topics/air-pollution. Accessed 20 Oct 2021
6. Castelli, M., Gonçalves, I., Trujillo, L., Popoviăź, A.: An evolutionary system for ozone concentration forecasting. Inf. Syst. Front. **19**(5), 1123–1132 (2017)
7. Corral-Plaza, D., Boubeta-Puig, J., Ortiz, G., García de Prado, A.: An Internet of things platform for air station remote sensing and smart monitoring. Comput. Syst. Sci. Eng. **35**(1), 5–12 (2020)
8. Díaz, G., Macià, H., Valero, V., Boubeta-Puig, J., Cuartero, F.: An intelligent transportation system to control air pollution and road traffic in cities integrating CEP and colored Petri nets. Neural Comput. Appl. **32**(2), 405–426 (2020)
9. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. In: 9th International Conference on Learning Representations, ICLR 2021, pp. 1–21 (2021)
10. Fakir, M.H., Kim, J.K.: Prediction of individual thermal sensation from exhaled breath temperature using a smart face mask. Build. Environ. **207**, 108507 (2022)
11. Grandini, M., Bagli, E., Visani, G.: Metrics for multi-class classification: an overview (2020). https://arxiv.org/abs/2008.05756
12. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York (2009). https://doi.org/10.1007/978-0-387-84858-7
13. Jato-Espino, D., Castillo-Lopez, E., Rodriguez-Hernandez, J., Ballester-Muñoz, F.: Air quality modelling in Catalonia from a combination of solar radiation, surface reflectance and elevation. Sci. Total Environ. **624**, 189–200 (2018)
14. Laubscher, R., Rousseau, P.: An integrated approach to predict scalar fields of a simulated turbulent jet diffusion flame using multiple fully connected variational autoencoders and MLP networks. Appl. Soft Comput. **101**, 107074 (2021)
15. Li, J., Shao, X., Zhao, H.: An online method based on random forest for air pollutant concentration forecasting. In: 37th Chinese Control Conference, CCC 2018, pp. 9641–9648. IEEE (2018)
16. Liu, T., Lau, A.K.H., Sandbrink, K., Fung, J.C.H.: Time series forecasting of air quality based on regional numerical modeling in Hong Kong. J. Geophys. Res. Atmos. **123**(8), 4175–4196 (2018)

17. Middya, A.I., Roy, S., Chattopadhyay, D.: CityLightSense: a participatory sensing-based system for monitoring and mapping of illumination levels. ACM Trans. Spat. Algorithms Syst. **8**(1), Article 5 (2021)
18. Palak, R., Wojtkiewicz, K., Merayo, M.G.: An implementation of formal framework for collective systems in air pollution prediction system. In: Nguyen, N.T., Iliadis, L., Maglogiannis, I., Trawiński, B. (eds.) ICCCI 2021. LNCS (LNAI), vol. 12876, pp. 508–520. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88081-1_38
19. Paoli, C., Notton, G., Nivet, M.-L., Padovani, M., Savelli, J.-L.: A neural network model forecasting for prediction of hourly ozone concentration in Corsica. In: 10th International Conference on Environment and Electrical Engineering, EEEIC 2011, pp. 1–4. IEEE (2011)
20. Semlali, B.B., El Amrani, C., Ortiz, G., Boubeta-Puig, J., García de Prado, A.: SAT-CEP-monitor: an air quality monitoring software architecture combining complex event processing with satellite remote sensing. Comput. Electr. Eng. **93**, 107257 (2021)
21. Seng, D., Zhang, Q., Zhang, X., Chen, G., Chen, X.: Spatiotemporal prediction of air quality based on LSTM neural network. Alex. Eng. J. **60**(2), 2021–2032 (2021)
22. Vaswani, A., et al.: Attention is all you need. In: 31st Conference on Neural Information Processing Systems, NIPS 2017, pp. 1–11 (2017)
23. Wu, N., Green, B., Ben, X., O'Banion, S.: Deep transformer models for time series forecasting: the influenza prevalence case (2020). https://arxiv.org/abs/2001.08317
24. Yu, Y., Si, X., Hu, C., Zhang, J.: A review of recurrent neural networks: LSTM cells and network architectures. Neural Comput. **31**(7), 1235–1270 (2019)