ISSN: **1992-8645** www.jatit.org E-ISSN: **1817-3195**

DEEP LEARNING AND TIME SERIES ANALYSIS APPLICATION ON TRAFFIC FLOW FORECASTING

[1]NADIA SLIMANI, [1]MUSTAPHA AMGHAR, [1]NAWAL SBITI

[1]SIP Research Team, Rabat IT Center, EMI, Mohammed V University, Rabat, Morocco

E-mail: slimani.nadia@gmail.com , amghar@emi.ac.ma , sbiti@emi.ac.ma

ABSTRACT

The use of new information and communication technologies is an important aid to solving transportation problems. This is commonly known as ITS (Intelligent Transport Systems) that can provide effective information for travelers and traffic managers. Road traffic prediction has led to a growing research area debated by several researchers affiliated to a range of disciplines. Recently, a significant amount of research efforts has been devoted to deep learning methods, greatly advancing traffic prediction abilities. The purpose of this paper is to meet a practical need: forecasting highway traffic volume by payment method (manual/ electronic toll) over a long time horizon based on historical observations. A variety of methods were used to make this forecast, and the results were compared and interpreted to identify model limitations and improve the approach. We propose to forecast daily highway traffic using four methods: Seasonal Autoregressive Integrated Moving Average with additional lagged values (SARIMAX), Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN) and at least we perform combination method which has been shown to be efficient in the literature and which, applied to our real case study, will confirm these performances. Indeed, the best prediction results were obtained using the hybridization CNN-LSTM which reached an R-squared value of 95%. In the present study, such an effort resulted in considerable improvement in long terme forecasting accuracy when compared with the existing models' performance previously adopted in the field of traffic forecasting.

Keywords: *Intelligent Transport Systems, Prediction Models, Deep Learning, Time Series Forecast, SARIMAX, Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN).*

1. INTRODUCTION

Road and highway infrastructure design process relies in a significant way on traffic forecasting. This phase is an important step in the investment feasibility study, as it allows to decide on the appropriateness of the investment by determining the transport demand, the predictive origin-destination matrix, and the distribution of cars in the different sections. These deliverables are made based on a set of interrelated factors and assumptions.

Traffic forecasting is an essential activity for the services in charge of traffic management and user information. The principle motivation behind this activity is to know the level of traffic and its evolution to program investments over time, determine the proper sizing of the pavement structure by knowing the level of heavy vehicle traffic, anticipate the degree of damping of the pavement to plan its maintenance according to the wear, and as a result, raise the quality of road infrastructure, which has a positive impact on user safety indicators. These studies also allow road infrastructure managers to ensure better road viability by allocating adequate resources and taking these factors into account, they can estimate appropriate pricing for toll road sections. Traffic forecasts are likewise utilized for some, different purposes including hallway arranging, frameworks arranging, interchange design, air quality investigation, and numerous exceptional ventures.

The objectives and challenges of traffic forecasting are different depending on the forecast horizon and the context of the study. During the recent decades, this issue was treated widely by several authors belonging to several specialties and with multiple approaches. As a matter of fact, those

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

works can be grouped under concentrating on procedures of traffic modeling or evaluating the performance of different prediction models including parametric and non-parametric techniques, or contrasting the viability of those models by looking at the forecast horizon (short, medium or long-term) and the context of study (urban or non-urban areas).

Traffic flow prediction relies heavily on historical and real-time traffic data collected from various sensor sources, including electromagnetic loops, radar, cameras, etc. Although many traffic flow prediction systems and models already exist, most of them use shallow traffic models and are still somewhat unsatisfactory. This prompts us to rethink the traffic flow prediction problem based on deep architecture models. Recently, deep learning, which is a type of machine learning method, has attracted much academic and industrial interest. It has been successfully applied in classification tasks, natural language processing, object detection, motion modeling, etc.

Deep learning algorithms use multi-layer or deep architectures to extract the inherent features of data from the lowest to the highest level. Since the traffic flow process is complicated in nature, deep learning algorithms can represent traffic features without prior knowledge, which provides good performance for traffic flow prediction.

The current research is in line with this perspective and it's a continuation of our previous works. The overall objective of our project is to predict the number of vehicles passing through each toll station per hour by taking into account factors that affect the traffic. The aim is to allow the managers, through these predictions, to manage and control the traffic in order to reduce congestion and consequently increase safety on the roads used, anticipate the sizing of the infrastructures and the resources to be allocated (reduce or increase the number of lanes and toll collectors).

The approach consists of traffic flow prediction based on a fairly consistent history and containing real daily traffic stream recorded by a recognized traffic manager in Morocco in order to predict the future traffic data using four forecasting models:

- Parametric models:
  - Seasonal Autoregressive Integrated Moving Average with additional lagged values (SARIMAX).
- Non-Parametric models:
  - Recurrent Neural Networks LSTM;
  - Convolutional Neural Networks CNN.
- Hybrid models:
  - CNN-LSTM.

In order to verify the effectiveness of the proposed model, two groups of datasets and different models are studied in the experiment. The performance of these models will then be compared between different simulations according to established criteria.

The current paper is sorted out as follows: The first section is dedicated to the literature review in which various strategies used to forecast traffic will be presented, the second section will cover data exploration and scope including the study's context. The subsequent part is dedicated to the proposed methodology by depicting in subtleties the dataset and the model of forecasting. Finally, in the fourth part, numerical experiments will be handled in order to compare the performances of the studied models according to predefined criteria alongside the conclusions.

2. LITTERATURE REVIEW

The literature review for this work comprises of the study of available literature on the methods previously used for traffic forecasting, their challenges, scope for improvement and then the study of the latest advances in the field, especially with reference to time-series analysis and artificial intelligence in the Moroccan context.

Traffic forecasting can be displayed as a time series regression problem to which we can apply the toolbox for time series analysis that consist in recording and measuring the same phenomenon over time.

2.1 (S)ARIMA(X)

Auto regressive integrated moving average (ARIMA) is a widely used model for time series analysis and has been applied to traffic forecasting (web traffic [1] and road traffic [2] [3]). ARIMA consists of three parts:

i) the Auto-regressive (AR) part indicates that the evolving variable of interest can be

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

approximated using a linear combination of its own historical values:

$$y_t = \mu + \sum_1^p \emptyset_i y_{t-i} + \varepsilon_t \quad (1)$$

With

$\emptyset_i$: The auto-regression coefficients.

Literally, each observation consists of a random component (random shock, ε_t) and a linear combination of the previous observations.

ii) the Moving average (MA) part is used to model the residual from the AR part using a weighted combination of random noises at various previous time steps. Moving average models suggest that the series has fluctuations around a mean value. The best estimate is then considered to be the weighted average of a number of previous values (which is the principle of the moving average procedures used for data smoothing). This amounts to considering that the estimate is equal to the true average, to which we add a weighted sum of the errors that have affected the previous values:

$$y_t = \mu + \sum_1^p \theta_i \varepsilon_{t-i} + \varepsilon_t \quad (2)$$

With

θ_i: The auto-regression coefficients.

Each observation is composed of a random error component (random shock, ε_t) and a linear combination of past random errors.

iii) the Integrate (I) part models the difference between adjacent values rather than raw values.

ARIMA is a mainstream model for time series investigation and has been applied to traffic forecasting. In our previous work [4], Seasonal ARIMA was used to catch the periodicity of traffic flow in different days and rush hours. In [5], Chai used a sparse seasonal ARIMA which can correctly predict the air traffic from January to July in 2020 with 95% confidence interval and can identify financial crisis, political storm, and the COVID-19 outbreak, ARIMA could be associated to other tools to increase the precision of the forecast by hybrid models [6].

Other mainstream techniques for traffic estimating incorporate K-Nearest Neighbor (KNN) [7] [8], Support Vector Regression (SVR) [9] [10], Hidden Markov Model[11] and Gaussian Process [12]. Time series analysis was used by Jha et al. [13] by underlining the lower values of estimation errors found with this method when compared to trend line analysis and econometric regression analysis. These methods usually pursue strict mathematical deduction and definite physical meaning, which are limited to less complex traffic conditions and relatively small size of traffic data.

2.2 Recurrent Neural Network (RNN)

To display the non-parametric component in traffic analysis, artificial intelligence based methodologies have likewise been applied to traffic determining. There are so many different neural networks that it is simply impossible to mention them all. But for our case, we will be particularly interested in two types of neural networks: Recurrent neural networks and Convolutional neural networks In [14], the authors propose to utilize neural networks to demonstrate the ability of forecasting with a good accuracy. The most recent methods are deep learning, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), where CNNs are usually used for spatial structure learning [15], while RNNs (e.g., LSTM [16] and GRU [17]) are widely used for temporal and sequential learning. In [18] [19] [20], perform traffic forecasts with deep learning, which is a kind of neural organization with self-association, and can perform nonlinear auto-relapse. Be that as it may, most of the previously mentioned approaches model each traffic time series independently, neglecting to catch the spatial reliance among them.

Recurrent neural networks (RNN) are networks with loops (Figure 1) that allow the information to persist [21].

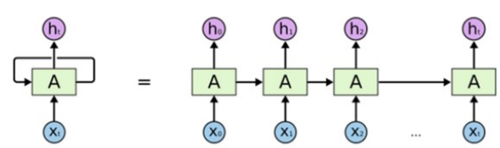


Figure 1: Loops Of RNN

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In the diagram of figure 1, a recurrent neural network is based on an input x_t to provide a value h_t. A loop allows to pass information from one step of the network to the other. A recurrent neural network can be considered as a set of copies of the same network, each transmitting a message to a successor, as shown in Figure 1.
The LSTM network is a special type of recurrent neural network, capable of learning long-term dependencies. It was introduced by Hochreiter & Schmidhuber [22] in 1997 and has been refined and popularized in many researches. LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their main advantage.

All recurrent neural networks have the form of a chain of repeating neural network modules. In standard RNNs, this repeating module will have a very simple structure, such as a single hyperbolic tangent layer. LSTM also have this chain structure, but the repeating module has a different structure [23]. Instead of having a single layer of neural network, there are four, interacting in a special way (figure 2). The various symbols shown in Figure 2 are explained in Figure 3.

Steps in an LSTM:

The first step in LSTM is to decide what information we are going to remove from the cell state. This decision is made by a sigmoid layer called the forget gate layer, which looks at h_{t-1} and x_t, and generates a number between 0 and 1 for each component of the cell state C_{t-1}. The value 1 represents "completely keep this" while a 0 represents "completely get rid of this" (figure 4).

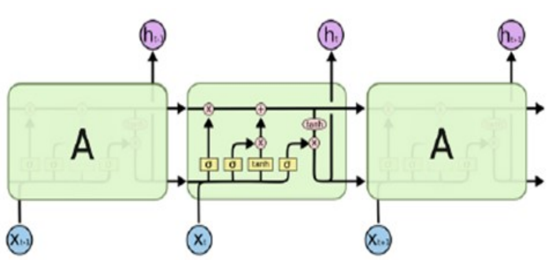


Figure 2: LSTM Structure

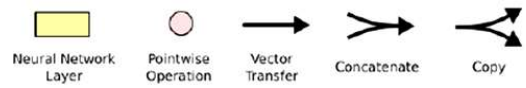
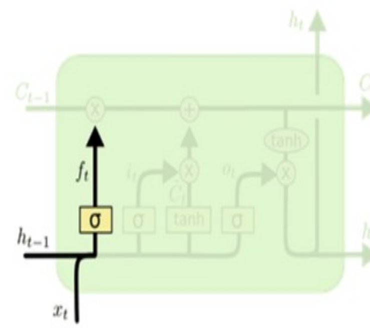


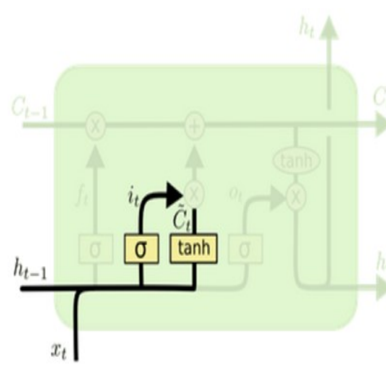
Figure 3: Symbols Used In LSTM Structure



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

Figure 4: LSTM Construction Step 1

The next step is to decide what new information we will store in the cell state. First, a sigmoid layer called input gate layer decides which values we will update. Then, a tanh layer creates a vector of new candidate values $\tilde{C}_t$, which could be added to the cell state. In the next step, we will combine these two to update the cell state (figure 5).

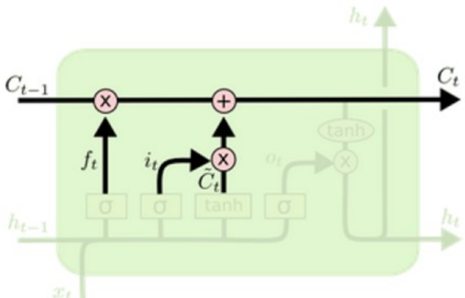


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Figure 5: LSTM Construction Step 2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The next step is to update the old cell state C_{t-1}. For this purpose, the old state is multiplied by f_t, forgetting the rest of the information. Then, the value $i_t * \widetilde{C}_t$ is added (figure 6).



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Figure 6: LSTM Construction Step 3

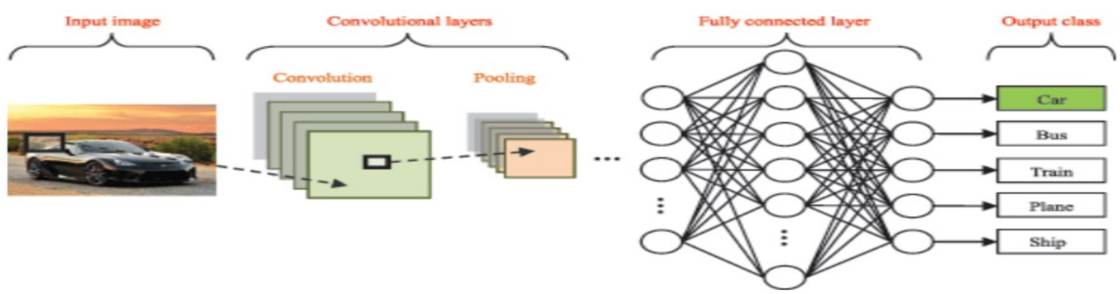
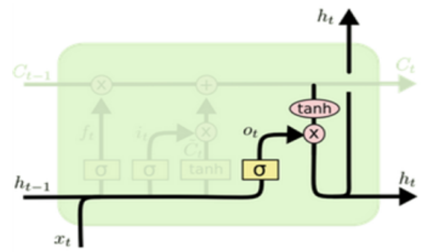


Figure 7: Illustration Of The CNN Architecture

Finally, we need to decide on the output of the network. This output will be based on our cell state, but after applying a filter. A sigmoid layer determines which components of the cell state we will let out. Then we pass the cell state values through a tanh layer (to put the values between -1 and 1) and multiply them by the output of the sigmoid gate [24], so that we only let out the information that we have decided to let through (figure 7).



$$o_t = \sigma\left(W_o \left[h_{t-1}, x_t\right] + b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Figure 8: LSTM Construction Step 4

2.3 Convolutional Neural Network (CNN)

CNNs are a class of deep neural networks capable of recognizing and classifying particular features of images and are widely used for visual image analysis [25]. Their applications range from image and video recognition [26] to image classification, medical image analysis [27],

An image is fed directly into the network, followed by several convolution and pooling steps. Then, the representations of these operations feed one or more fully connected layers. Finally, the last fully connected layer provides the class label.

This network architecture is inspired by the functioning of the visual cortex of animals. The analysis of the visual field is done through a set of overlapping sub-regions that pave the image. Each sub-region is analyzed by a neuron of the animal's brain, in order to pre-process small amounts of information. This is called convolutional processing.

The architecture of a convolutional neural network is formed by a succession of processing blocks to extract the features that discriminate the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

class to which the image belongs from others [31]. A processing block consists of one or more:

- Convolution layers (CONV) that process the data of a receiving field;
- Correction layers (ReLU), often called by abuse "ReLU" in reference to the activation function (Linear Rectification Unit);
- Pooling layers (POOL), which allows to compress the information by reducing the size of the intermediate image (often by subsampling).

The processing blocks follow each other to the final layers of the network, which perform the image classification and the calculation of the error between the prediction and the target value:

- "Fully Connected" (FC) layer, which is a perceptron-like layer;
- Loss Layer (LOSS).

2.3.1 Convolutional layer

This layer is the first layer that is used to extract different features from the input images. In this layer, the mathematical operation of convolution is performed between the input image and a filter of a particular size MxM. By dragging the filter over the input image, the scalar product is taken between the filter and the parts of the input image with respect to the filter size (MxM) [32].

The output is called a feature map which gives us information about the image such as corners and edges. Later, this feature map is fed into other layers to learn several other features of the input image.

2.3.2 Pooling layer

The pooling step is a subsampling technique [33]. Typically, a pooling layer is inserted regularly between the correction and convolution layers. By reducing the size of the feature maps (figure 9), and thus the number of network parameters, this speeds up the computation time and reduces the risk of overlearning.

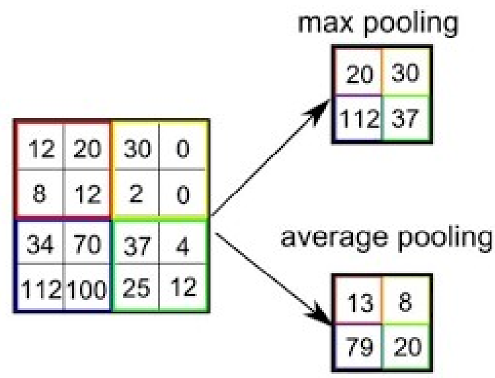


Figure 9: Scheme Of A Pooling Operation

2.3.3 Fully connected layer

This layer is at the end of the network. It allows the classification of the image from the features extracted by the succession of processing blocks. It is fully connected, because all the inputs of the layer are connected to the output neurons of this layer. They have access to all the input information. Each neuron assigns to the image a probability value of belonging to class i among the N possible classes [34].

2.3.4 Dropout

Usually, when all features are connected to the fully connected layer, this can lead to overfitting in the training dataset. Overfitting occurs when a particular model works so well on the training data that it negatively impacts the performance of the model when used on new data.

To overcome this problem, an exclusion layer is used in which a few neurons are removed from the neural network during the training process [35], thereby reducing the size of the model. When the exclusion layer is equal to 30% of the nodes are randomly removed from the neural network.

3. SCOPE AND DATA EXPLORATION

After defining the several approaches and methods used in the literature and our previous works for road traffic forecasting, the aim of this section is to provide hourly traffic forecasting with multivariate and multi-output series (for the transactions passed in electronic toll or manual payment) with a numerical simulation in a real context.

3.1 Setting and context of the study

Our study is conducted on the eastern section of the Trans-Maghreb highway, this section is 475 km long and connects the cities of Sidi Allal Bahraoui, Fez, Taza and Oujda with an average travel time of 3h58min.

3.2 Description of data sources

The work of building the predictive models is based on records between 2016 and 2019 made on a specific freeway's axis.

The database gathers the hourly traffic at each station of the studied axis, this flow is grouped by vehicle class and by payment method, as shown in the table 1.

Vehicles are classified into three categories, the criteria used to define the toll classes are physical and measurable:

- The overall height (including windshield) at the front axle of the vehicle or combination;
- The number of axles in the vehicle or combination;
- The length of the vehicle.

The vehicle passage data has three classes:

- CL1: passenger vehicles;
- CL2: vans;
- CL3: semi-trailers with 3 or more axles.

Table 1: Overview Of Transactional Traffic Data.

Station	Day	Hour	TELEP_CL1	CARD_CL1	CC_CL1	CASH_CL1	TELEP_CL2
Tiflet	23/08/2016	6	0	0	0	8	10
Taza Est	28/06/2016	5	0	1	0	10	4
Tahla	06/05/2016	19	0	0	0	43	7

The payment method represents the way of paying the highway fees, the payment methods are divided into two categories:

- Manual: the payment is made manually either by credit card or by cash and in this case the customer goes through a manual toll lane (figure 10) to make a transaction either by freeway subscription card (column 'CARD'), or by credit card (column 'CC') or by cash (column 'CASH').
- Electronic toll collection: the transaction is done automatically by electronic toll collection badges (figure 11); in this case the customer does not need to stop or enter a queue to pay the freeway fees (TELEP);

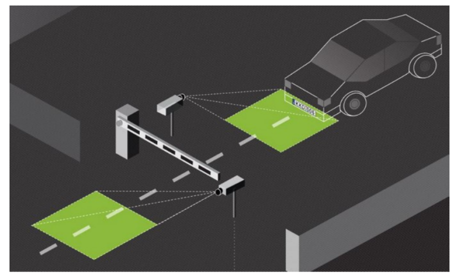


Figure 10: Detection By Non-Intrusive Sensor

Figure 11: Detection By Electronic Tolls

3.3 Data processing

In order to prepare the data for the analysis and learning phase, we will first proceed to a join and grouping of the data received in several batches. Then, a treatment is carried out to keep only those which will be useful.

The database will be subjected to several simplification and aggregation treatments of the means of payment according to time. After the data processing we will extract only the data related to

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the Sidi Allal El Bahraoui station (station with the biggest traffic flow). The resulting database of these treatments is illustrated below:

Table 2 : Aggregated Database Used For Model Analysis.

datetime	Telepeage	Manuel	TOTAL
2017-04-17 14:00:00	9	468	471
2017-04-17 15:00:00	12	447	451
2017-04-17 16:00:00	39	446	461
2017-04-17 17:00:00	102	545	579
2017-04-17 18:00:00	84	483	511

3.4 Data visualization

Once we are sure that the dataset is complete and does not contain erroneous values, we proceed to the visualization.

```
DatetimeIndex: 35017 entries, 2016-01-01 00:00:00 to 2019-12-31 23:00:00
Data columns (total 3 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Telepeage  35017 non-null  int64
 1   Manuel     35017 non-null  int64
 2   TOTAL      35017 non-null  int64
dtypes: int64(3)
memory usage: 1.1 MB
```

In a first step we visualize the hourly variation of traffic to know its hourly behavior, the figure 12 presents on the x-axis the evolution in time as well as on the y-axis the numbers of vehicles passing through the toll center at each hour.

We notice that the hourly presentation of the series does not allow us to extract much information, for this reason we opt in a second time to present in figure 12 below the daily, weekly and monthly evolution of the series to allow us to understand and extract the most information about the traffic.

For a clearer visualization of the traffic flow variations, we will zoom on the daily evolution of each year to be able to analyse and interpret the reasons causing exceptional variations

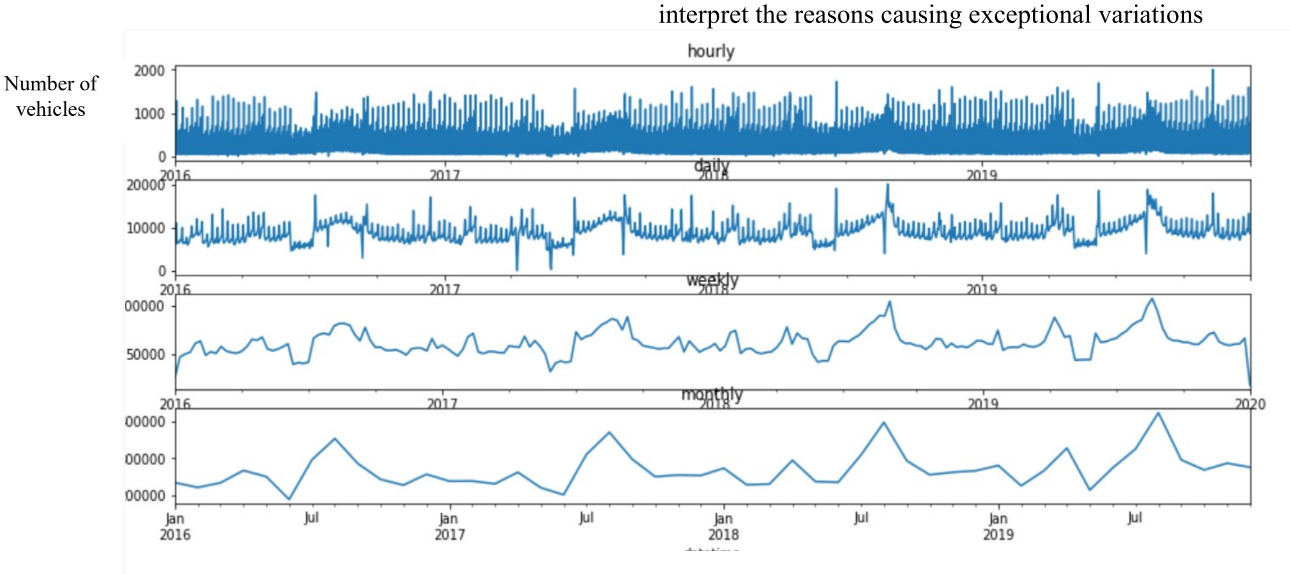


Figure 12: Breakdown Of Traffic Evolution Over The Different Periods At The Studied Station

The database used in this study is a time series that describes the evolution of traffic as a function of time over 4 years dated from January 2016 to December 2019, consists of 3 columns and 35017 rows.

After a historical analysis of the traffic data presented in figure 4, we can see that the traffic is not a random event but is directly linked to the vacations and public holidays of the year, according to the graph in figure13 below we notice that the

Journal of Theoretical and Applied Information Technology
15th March 2022. Vol.100. No 5
2022 Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

evolution of traffic is impacted by six events that must be retained when applying the models:

- Weekends: there is peaks at each weekend in Sundays;
- Ramadan period: the traffic undergoes a decrease during the Ramadan period;
- School vacations longer than 15 days: significant increase in traffic;
- Religious celebration: Decrease on the day of Eid followed by an increase the days after;
- Summer vacations: constant increase from the first days.

Before applying the predictive models, we will need to complete our database with the calendar variables that influence the traffic, so that if the date corresponds with one of the dates of these variables it will be associated with the value "1" otherwise "0". The table shows an extract of the database after adding the external variables

Table 3 : Extract of the database completed by the exogenous variables.

Day	Sunday	Ramadan	Eid	School Vacation	Summer Vacation
01/01/2016	0	0	0	1	0
02/01/2016	0	0	0	0	0
03/01/2016	1	0	0	0	0
04/01/2016	0	0	0	0	0

4. PROPOSED METHODOLOGY

4.1 Statistic model SARIMAX:

ARIMA and SARIMA are excellent tools for time series analysis. ARIMA stands for autoregressive integrated moving average. SARIMAX is similar and stands for seasonal autoregressive integrated moving average with exogenous factors.

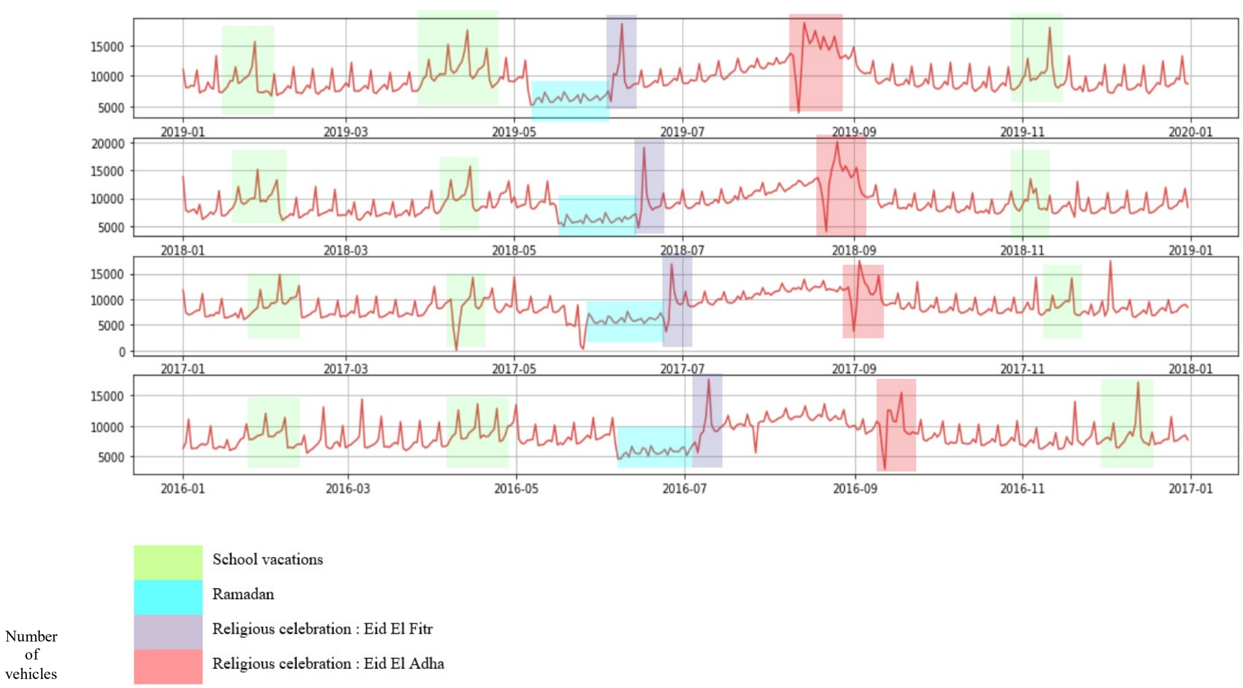


Figure 13 : Daily Traffic Patterns And Impact Factors

SARIMA accepts an additional set of parameters $(P, D, Q)\ m$ that specifically describe the seasonal components of the model. Here, P, D, and Q represent the seasonal regression, differentiation, and moving average coefficients, and m represents the period of a season in the series:

P: Seasonal autoregressive order.
D: Seasonal difference order
Q: Seasonal moving average order
m: The number of time steps for a single seasonal period.
The notation is written as (p,d,q)×(P,D,Q)m.

The objective of this modelling is to predict the total number of vehicles per day. Taking into account the exogenous variables.

The data set contains 1461 days: between January 2016 and December 2019.

4.1.1 Seasonality and trend

To better understand the nature of the series and to determine its seasonality and the nature of its trend, a temporal decomposition of the series is performed (Figure 14).

We can see that there is a clear seasonality in the data and the trend is both increasing and decreasing due to the fact that the number of vehicles may be higher on weekends than on weekdays. Thus m = 7 days.

4.1.2 Stationarity test

Before applying the SARIMA model we have to make sure that the data are stationary, for this we can perform the Dicky Fuller test [36] (Figure 15). The p-value obtained is greater than 0.05, so the time series is not stationary [37].

```
Strong evidence against Null Hypothesis
Reject Null Hypothesis - Data is Stationary
Results of Dickey-Fuller Test:
Test Statistic                 -0.752604
p-value                         0.390447
Lag used                       23.000000
Number of observation        1437.000000
Critical Value (1%)            -2.567298
Critical Value (5%)            -1.941189
Critical Value (10%)           -1.616636
dtype: float64
Strong evidence for  Null Hypothesis
Accept Null Hypothesis - Data is not Stationary
Data is NOT Stationary for traffic flow
```

Figure 15 : Dicky Fuller Test

In order to make the time series stationary, we will proceed to differentiations until it becomes approximately stationary. So after differentiating the series once, we run the Dicky Fuller test again.

The result is $P-value - 2.398 \times 10^{-13} < 0.05$
So the series became stationary after a single order of differentiation. For this purpose, the parameters D =d = 1.

4.1.3 Autocorrelation

The visualization of the autocorrelation, in other words, the correlation between the series and its lags, is done through the ACF graph, which also allows to estimate the other orders of the model, the figure 16 presents the ACF and PACF graphs [38].

From the ACF and PACF graphs, we notice a strong autocorrelation between the pitch and its second order shifts.

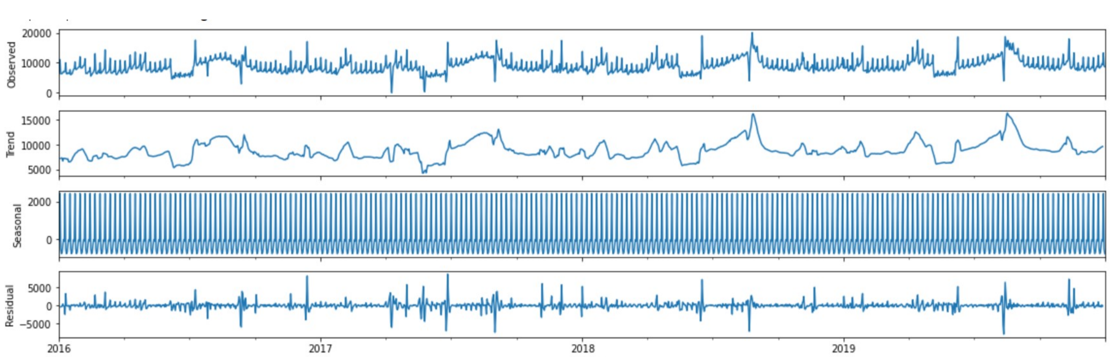


Figure 14 : Decomposition Of The Time Series

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

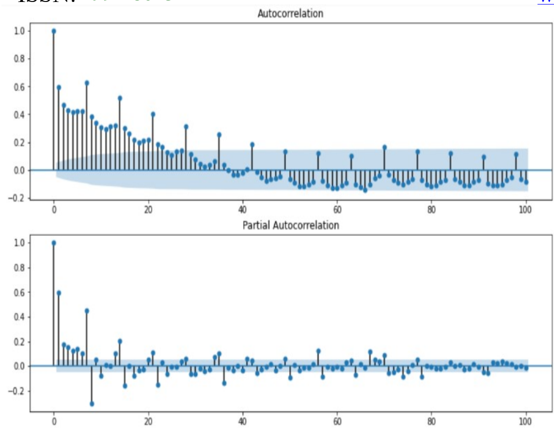


Figure 16 : ACF and PACF Test

4.1.4 The Akaike Information Critera (AIC)

The Akaike Information Criterion (AIC) is a mathematical method for evaluating how well a model fits the data from which it was generated [39]. In statistics, the AIC is used to compare different possible models and determine which one best fits the data. The AIC is calculated from:

- The number of independent variables used to build the model.
- How well the model reproduces the data.

The best-fitting model according to the AIC is the one that explains the most variation using the fewest independent variables [40]. The AIC function is used to find the optimal parameters for the SARIMA model. This provides SARIMAX seasonal order of (2, 1, 2) (1,0,1)7.

Train set: This one will be the most voluminous in terms of data. Indeed, it is on this set that the network will iterate during the training phase to be able to appropriate parameters, and adjust them as well as possible. Some rules recommend that it be composed of ¾ of the available data. This is the learning phase.

Test set: The role of this last one is to evaluate the network in its final form, and to see how well it predicts as if the network were integrated into our application. This is why it must be composed exclusively of new samples, never used before, to avoid biasing the results by sending it data that it already knows and that it has already learned during the training or validation phase. This one still can be estimated of the order of ¼ of the available data.

4.1.6 Performance criterion

In order to analyze the performance of each traffic forecasting method, we used three indexes of performance to evaluate the model's performance:

- Mean Absolute Error (MAE);
- Root Mean Square Error (RMSE);
- R-squared (R^2).

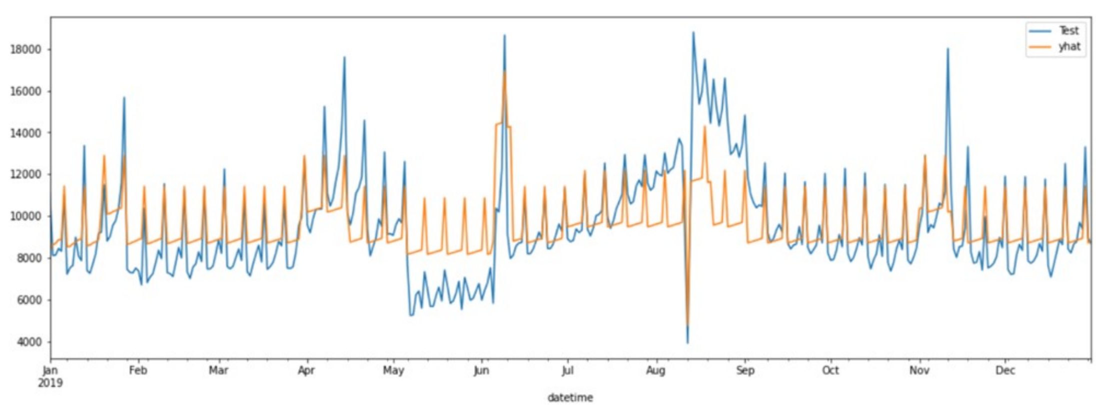


Figure 17 : Forecasting Results By SARIMAX Model

4.1.5 Train and test set

Before applying the model, we need to divide our database into train/test base sets.

Mean Absolute Error (MAE) is a measure of errors between paired observations expressing the same phenomenon. MAE is an arithmetic average of the absolute errors calculated as:

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}$$

With: $e_i = y_i - x_i$

When:

y_i are the predicted values

x_i are the recorded values

Root Mean Square Error (RMSE) is the square root of the MSE (Mean Square Error) index. MSE is defined as the arithmetic mean of the squares of the differences between model forecasts and observations. RMSE is calculated as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (|y_i - x_i|)^2}{n}}$$

R-squared (R²) is called also the coefficient of determination. It is an indicator used in statistics to judge the quality of a linear regression. Mathematically, it is the proportion of the variance of a dependent variable that is explained by one or more independent variables in the regression model. It is expressed either between 0 and 1, or as a percentage. While the correlation explains the strength of the relationship between an independent variable and a dependent variable, the R-squared explains the extent to which the variance of one variable explains the variance of the second variable.

$$R^2 = \frac{Variance\ explained\ by\ the\ model}{Total\ variance}$$

4.1.7 The results of experimentation

After having determined the orders of the SARIMAX model, we will proceed to its adjustment. For that, we used the package stats mode available on Python, the figure 17 presents the results of the model.

The performances obtained from the model are:

R^2=45% RMSE=1806.18 MAE=1280.46

From the results obtained, we notice that the model produces an error between 1800 and 1300 vehicles on each 2000 detected with a percentage of accuracy of 45%. An accuracy far from the blurred performance. The limitation lies in the presence of several external variables not taken into account. To overcome these limitations, deep learning models will be used in the next part.

4.2 Recurrent Neural Networks: LSTM

In this section we will focus on hourly traffic forecasting with multi-output series (electronic tolls, manual). The deep learning models require a specific format of the input data, they must be three dimensional: (number of sequences, number of steps in time, number of variables). As the previous part, we divide our

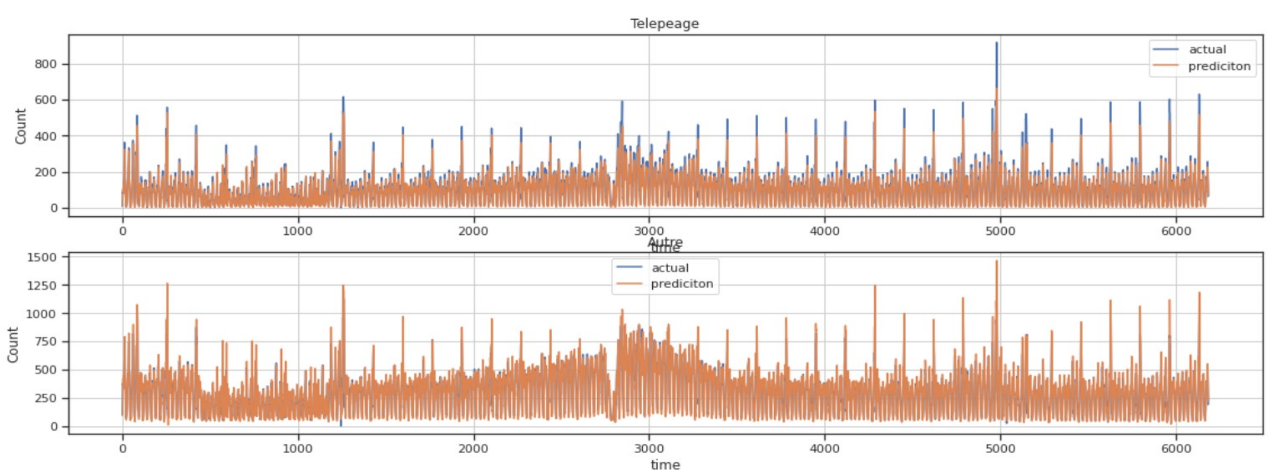


Figure 18 : Forecasting Results By LSTM Model

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

database in two parts: " train set " and " Test set".

LSTM recurrent neural networks are useful for modeling time series data because the network maintains a memory, learning to retain useful information from the inputs of previous model inferences. Each time the model is called, the memory is updated based on the updated inputs.

4.2.1 Structure of the LSTM model

In order to build an LSTM model, we need to import some modules from Keras:
Sequential: to initialize the neural network
LSTM: to add the long term memory layer.
Dense: to add a densely connected neural network layer.
Dropout: to add exclusion layers that prevent overfitting.

When defining the Dropout layers, we specify 0,1, which means that 10% of the layers will be dropped. Then we add the Dense layer which specifies the output of 2 units.

4.2.2 Training the LSTM model

The LSTM model will be trained with two layers containing 124 hidden units each, with RELU as the activation function. Then, the popular optimizer ADAM will be used to define the loss as the mean absolute deviation. Finally, the model will be trained with 16 sequences per batch for 50 epochs.

4.2.3 LSTM results

After the construction and training phase we test our model on the test set to determine the model performances showed in table 4 bellow. The results obtained are illustrated in figure 18.

Table 4 : The Recorded Performance Of LSTM Model.

Payment method	*R^2*	*RMSE*	*MAE*
Manual	*90.02 %*	*60.68*	*40.70*
Electronic toll	*79.02 %*	*36.49*	*23.13*

In spite of the performance of the LSTM recurrent neural networks, they present a major disadvantage that influences the quality of the results, in fact, they have difficulty in treating the time series that present several variations in the time. For this we will try convolutional neural networks that are able to identify and extract changes in the series.

4.3 Convolutional Neural Networks: CNN

The third prediction method is to use a 1D convolution model.

4.3.1 Structure of the 1D CONV model

In order to build a 1D convolutional model we need to import the sub-modules of the Keras library:
Sequential: to initialize the neural network
Conv1D: to add the convolutional layer.
Dense: to add a densely connected neural network layer.
Maxpooling: to reduce the size of the data.
Flaten: to convert the data into a one-dimensional array for transmission to the next layer.
Dropout: to add exclusion layers that prevent overfitting.

When defining the Dropout layers, this time we specify 0.15, which means that 15% of the layers will be dropped. Then we add the Dense layer that specifies the 2 unit output to identify to the program that it is a two vector output.

4.3.2 Training the CNN model

The 1D CONV network will be trained with a single layer containing 64 filters, 5 cores, with RELU as the activation function. Then, it will be compiled in the same way as in the previous part using the ADAM optimizer and we define the loss function as the mean absolute deviation. Finally, we train the model on the training dataset with 16 sequences per batch for 65 epochs.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

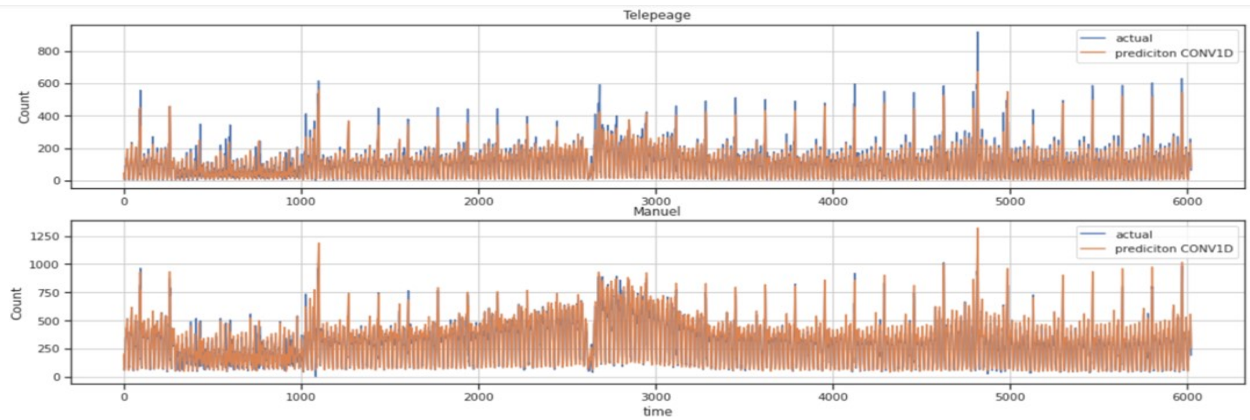


Figure 19 : Forecasting Results By CNN Model

4.3.3 CNN results

After the model building and training phase we test our model on the test set to determine the performance of the model. Figure 19 bellow presents the results of the CONV 1D model by comparing it with the real data. Also, the model performances are shown in table 5 bellow.

Table 5 : The Recorded Performance Of CNN Model.

Payment method	*R²*	*RMSE*	*MAE*
Manual	*95.06 %*	*40.29*	*28.21*
Electronic toll	*87.84 %*	*28.29*	*16.93*

The 1D convolution model gives more accurate results than the LSTM with a percentage of accuracy of 95.06% for the forecasts of the manual station users and 87.84% for the forecasts of the electronic toll users.

Although the convolutional model presents better results than the other models, it also presents some limitations that can negatively impact the quality of its forecasts, among the disadvantages that present themselves at the level of CNNs that this type of models can not learn the temporal ...rizes the time series.

Therefore, to solve the problem of time dependence for CNN and the limitation at the level of LSTM for the detection of features, a hybrid model that combines the two is proposed, the CNN-LSTM model.

4.4 Hybrid model: CNN-LSTM

A convolutional neural network is a network that applies a process called convolution to determine the relationships between two features. These networks are traditionally used for image classification and do not take into account sequential dependencies like a recurrent neural network can.

However, the main advantage of CNNs that makes them suitable for time series prediction is that of dilated convolutions - or the ability to use filters to calculate dilations between each cell. That is, the size of the space between each cell, which allows the neural network to better understand the relationships between different observations in the time series. For this reason, the LSTM and CNN layers are often combined when predicting a time series (see figure 20). This allows the LSTM layer to take into account sequential dependencies in the time series, while the CNN layer completes this process through the use of dilated convolutions. That's why, standalone CNNs are increasingly being used for time series forecasting, and the combination of multiple Conv1D layers can actually produce quite impressive results - rivaling those of a model that uses both CNN and LSTM layers.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

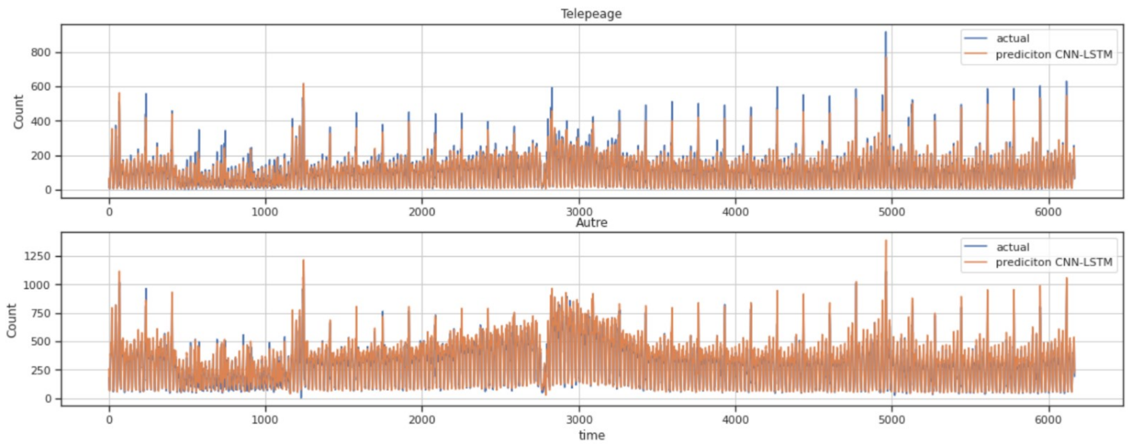


Figure 20 : Forecasting Results By CNN-LSTM Hybrid Model

4.4.1 CNN-LSTM structure

The data is first reshaped and resized to match the three-dimensional input requirements of the Keras sequence model. The input shape would be 24 time steps (Windows size = 24) with 10 features for a multivariate model with 2 outputs.

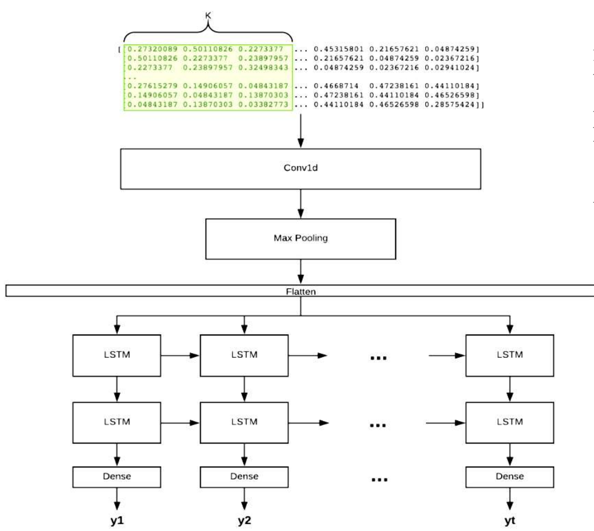


Figure 21 : CNN-LSTM Architecture For Time Series

We used a 1D convolutional layer followed by a Maxpooling layer in figure 21, the output is then flattened using a Flaten layer to feed the LSTM layers. The model has two hidden LSTM layers followed by a dense layer to provide the output.

The model uses the Adam RMS version of stochastic gradient descent for optimization and the mean absolute deviation as a function to calculate the error. Once the model is defined, it is fitted to the training data and the fitted model is used to make a prediction using the test data.

The performances obtained by the hybrid model in table 6 confirm our assumptions, and are beyond those recorded by the previous models. The performance of the model could not exceed 95% which is already a largely acceptable performance for forecasting, the remaining 5% of error is due to unforeseen changes in traffic such as accidents or due to other variables such as weather conditions, or local events such as soccer matches or local festivities. The accuracy of the results can be improved by taking into consideration other databases that describe the exogenous variables.

Table 6 : The Recorded Performance Of CN-LSTM Model.

Payment method	*R^2*	*RMSE*	*MAE*
Manual	*95.16%*	*40.81*	*27.67*
Electronic toll	*91.2%*	*24.79*	*15.01*

The experimental results show that the augmented model CNN-LSTM is better than the existing baseline methods SARIMAX and CNN and has more adaptability and higher accuracy in long-term traffic flow forecasting.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The experiment conducted to verify the effectiveness of the proposed model was limited in case of peaks due to generally to national events or specific meteorological conditions. In order to establish a more accurate prediction model, the entire factors that influence road network should be used as the research object. In future research, many influencing factors such as pavement works, traffic accidents, temperature, precipitation, sports or cultural events will be considered as inputs.

5. CONCLUSION

This study departed from the abundance of contributions dealing with techniques for road traffic forecasting. In this paper, we have discussed and employed several methods of machine learning algorithms and time series analysis to design an accurate model to predict daily road traffic data. The study uses a real data set of historical data from the past 3 years on eastern section of the Trans-Maghreb highway, Morocco. In the present study, such an effort resulted in considerable improvement in forecasting accuracy when compared with the existing models' performance previously adopted in the field of traffic forecasting.

The main contribution of our work is the development of four models of traffic prediction: CNN, LSTM, SARIMAX and a hybrid model called CNN-LSTM. Those models were, then, compared based on predefined criteria. The experimental results demonstrated that, according to the most commonly used criteria of error measurements (RMSE, MAE, and R^2), the hybrid model CNN-LSTM is performing better in terms of prediction accuracy and stability. The use of Akaike's Information Criterion technique (AIC) has also shown that the proposed model has a higher performance.

The proposed traffic prediction model represents a reliable tool for users and managers for:

- Maintaining a quality forecast at any time of the day or period of the year;
- Anticipation of phenomena that may disrupt the network;
- Multi-horizon anticipation of traffic conditions for intervention;
- Use of adapted predictive tools: maintenance of a forecast time compatible with the real time objectives of intervention and ease of forecasting on the scale of the network.

Despite the good performance recorded by the CNN-LSTM model, it remains perfectible in terms of traffic peak prediction. Thus, our next work will focus on a solution to integrate data from several sources: transactions, cameras, social networks... and in different formats (videos, images, texts) in order to improve the quality of our predictions whatever the traffic conditions.

REFRENCES:

[1] C. A. H. Suarez et O. J. S. Parra, « An ARIMA model for forecasting Wi-Fi data network traffic values », p. 5.

[2] X. Luo, L. Niu, et S. Zhang, « An Algorithm for Traffic Flow Prediction Based on Improved SARIMA and GA », *KSCE J. Civ. Eng.*, vol. 22, nº 10, p. 4107-4115, oct. 2018, doi: 10.1007/s12205-018-0429-4.

[3] N. Slimani, I. Slimani, N. Sbiti, et M. Amghar, « Machine Learning and statistic predictive modeling for road traffic flow », *Int. J. Traffic Transp. Manag.*, vol. 03, nº 01, p. 17-24, mars 2021, doi: 10.5383/JTTM.03.01.003.

[4] N. Slimani, I. Slimani, M. Amghar, et N. Sbiti, « Comparative analysis of traffic prediction models based on a real case study ». INTIS 2019.

[5] S. Chai, « Hong Kong Air Traffic: Explanation and Prediction based on Sparse Seasonal ARIMA Model », *ArXiv210805817 Stat*, août 2021, Consulté le: août 29, 2021. [En ligne]. Disponible sur: http://arxiv.org/abs/2108.05817

[6] M. Ge, Z. Junfeng, W. Jinfei, H. Huiting, S. Xinghua, et W. Hongye, « ARIMA-FSVR Hybrid Method for High-Speed Railway Passenger Traffic Forecasting », *Math. Probl. Eng.*, vol. 2021, p. e9961324, mai 2021, doi: 10.1155/2021/9961324.

[7] X. Luo, D. Li, Y. Yang, et S. Zhang, « Spatiotemporal Traffic Flow Prediction with KNN and LSTM », *J. Adv. Transp.*, vol. 2019, p. 1-10, févr. 2019, doi: 10.1155/2019/4145353.

[8] D. Xu, Y. Wang, P. Peng, S. Beilun, Z. Deng, et H. Guo, « Real-time road traffic state prediction based on kernel-KNN », *Transp. Transp. Sci.*, vol. 16, nº 1, p. 104-118, déc. 2020, doi: 10.1080/23249935.2018.1491073.

[9] N. Slimani, I. Slimani, M. Amghar, et N. Sbiti, « Road traffic forecasting using a real data set

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

in Morocco », *Procedia Comput. Sci.*, vol. 177, p. 128-135, 2020, doi: 10.1016/j.procs.2020.10.020.

[10] F. Zhou, Q. Yang, K. Zhang, G. Trajcevski, T. Zhong, et A. Khokhar, « Reinforced Spatiotemporal Attentive Graph Neural Networks for Traffic Forecasting », *IEEE Internet Things J.*, vol. 7, nº 7, p. 6414-6428, juill. 2020, doi: 10.1109/JIOT.2020.2974494.

[11] Y. Qi et S. Ishak, « A Hidden Markov Model for short term prediction of traffic conditions on freeways », *Transp. Res. Part C Emerg. Technol.*, vol. 43, p. 95-111, juin 2014, doi: 10.1016/j.trc.2014.02.007.

[12] A. Bayati, V. Asghari, K. Nguyen, et M. Cheriet, « Gaussian Process Regression Based Traffic Modeling and Prediction in High-Speed Networks », in *2016 IEEE Global Communications Conference (GLOBECOM)*, Washington, DC, USA, déc. 2016, p. 1-7. doi: 10.1109/GLOCOM.2016.7841857.

[13] K. Jha, N. Sinha, S. Arkatkar, et A. Sarkar, « A Comparative Study on Application of Time Series Analysis for Traffic Forecasting in India: Prospects and Limitations », *Curr. Sci.*, vol. 110, p. 373, févr. 2016, doi: 10.18520/cs/v110/i3/373-385.

[14] N. Slimani, I. Slimani, N. Sbiti, et M. Amghar, « Traffic forecasting in Morocco using artificial neural networks », *Procedia Comput. Sci.*, vol. 151, p. 471-476, 2019, doi: 10.1016/j.procs.2019.04.064.

[15] B. Yu, Y. Lee, et K. Sohn, « Forecasting road traffic speeds by considering area-wide spatio-temporal dependencies based on a graph convolutional neural network (GCN) », *Transp. Res. Part C Emerg. Technol.*, vol. 114, p. 189-204, mai 2020, doi: 10.1016/j.trc.2020.02.013.

[16] B. Yang, S. Sun, J. Li, X. Lin, et Y. Tian, « Traffic flow prediction using LSTM with feature enhancement », *Neurocomputing*, vol. 332, p. 320-327, mars 2019, doi: 10.1016/j.neucom.2018.12.016.

[17] M. Jiang, W. Chen, et X. Li, « S-GCN-GRU-NN: A novel hybrid model by combining a Spatiotemporal Graph Convolutional Network and a Gated Recurrent Units Neural Network for short-term traffic speed forecasting », *J. Data Inf. Manag.*, janv. 2021, doi: 10.1007/s42488-020-00037-9.

[18] M. X. Dai, M. R. Fu, M. Y. Lin, F.-Y. Wang, et L. Li, « DeepTrend: A Deep Hierarchical Neural Network for Traffic Flow Prediction », p. 6, 2017.

[19] P. Ø. Kanestrøm, « Traffic flow forecasting with deep learning », p. 96.

[20] Hongsuk Yi, HeeJin Jung, et Sanghoon Bae, « Deep Neural Networks for traffic flow prediction », in *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Jeju Island, South Korea, févr. 2017, p. 328-331. doi: 10.1109/BIGCOMP.2017.7881687.

[21] A. Kisvari, Z. Lin, et X. Liu, « Wind power forecasting – A data-driven method along with gated recurrent neural network », *Renew. Energy*, vol. 163, p. 1895-1909, janv. 2021, doi: 10.1016/j.renene.2020.10.119.

[22] S. Hochreiter et J. Schmidhuber, « LSTM can solve hard long time lag problems ». 1997.

[23] B.-J. Hou et Z.-H. Zhou, « Learning With Interpretable Structure From Gated RNN », *IEEE Trans. Neural Netw. Learn. Syst.*, p. 1-13, 2020, doi: 10.1109/TNNLS.2020.2967051.

[24] Y. Li, Y. He, et M. Zhang, « Prediction of Chinese energy structure based on Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) », *Energy Sci. Eng.*, vol. 8, nº 8, p. 2680-2689, août 2020, doi: 10.1002/ese3.698.

[25] J. A. de J. Osuna-Coutiño et J. Martinez-Carranza, « Structure extraction in urbanized aerial images from a single view using a CNN-based approach », *Int. J. Remote Sens.*, vol. 41, nº 21, p. 8256-8280, nov. 2020, doi: 10.1080/01431161.2020.1767821.

[26] O. Elharrouss, N. Almaadeed, S. Al-Maadeed, A. Bouridane, et A. Beghdadi, « A combined multiple action recognition and summarization for surveillance video sequences », *Appl. Intell.*, vol. 51, nº 2, p. 690-712, févr. 2021, doi: 10.1007/s10489-020-01823-z.

[27] K.-K. Tseng, R. Zhang, C.-M. Chen, et M. M. Hassan, « DNetUnet: a semi-supervised CNN of medical image segmentation for super-computing AI service », *J. Supercomput.*, vol. 77, nº 4, p. 3594-3615, avr. 2021, doi: 10.1007/s11227-020-03407-7.

[28] R. Yang *et al.*, « CNN-LSTM deep learning architecture for computer vision-based modal frequency detection », *Mech. Syst. Signal Process.*, vol. 144, p. 106885, oct. 2020, doi: 10.1016/j.ymssp.2020.106885.

[29] S. Ahlawat, A. Choudhary, A. Nayyar, S. Singh, et B. Yoon, « Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN) », *Sensors*, vol. 20, nº 12, p. 3344, juin 2020, doi: 10.3390/s20123344.

[30] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, et U. R. Acharya, « ABCDM: An Attention-based Bidirectional CNN-RNN Deep Model for sentiment analysis », *Future Gener. Comput. Syst.*, vol. 115, p. 279-294, févr. 2021, doi: 10.1016/j.future.2020.08.005.

[31] A. Ajit, K. Acharya, et A. Samanta, « A Review of Convolutional Neural Networks », in *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, Vellore, India, févr. 2020, p. 1-5. doi: 10.1109/ic-ETITE47903.2020.049.

[32] Z. J. Wang *et al.*, « CNN Explainer: Learning Convolutional Neural Networks with Interactive Visualization », *IEEE Trans. Vis. Comput. Graph.*, vol. 27, nº 2, p. 1396-1406, févr. 2021, doi: 10.1109/TVCG.2020.3030418.

[33] M. Momeny *et al.*, « A Convolutional Neural Network based on Adaptive Pooling for Classification of Noisy Images », *Signal Data Process.*, vol. 17, nº 4, p. 139-154, févr. 2021, doi: 10.29252/jsdp.17.4.139.

[34] S. H. S. Basha, S. R. Dubey, V. Pulabaigari, et S. Mukherjee, « Impact of fully connected layers on performance of convolutional neural networks for image classification », *Neurocomputing*, vol. 378, p. 112-119, févr. 2020, doi: 10.1016/j.neucom.2019.10.008.

[35] G. S. Nandini, A. P. S. Kumar, et C. K, « Dropout technique for image classification based on extreme learning machine », *Glob. Transit. Proc.*, vol. 2, nº 1, p. 111-116, juin 2021, doi: 10.1016/j.gltp.2021.01.015.

[36] S. Kumar et N. Kumar, « Testing normality in the time series of EMP indices: an application and power-comparison of alternative tests », *Commun. Stat. - Theory Methods*, p. 1-14, mai 2021, doi: 10.1080/03610926.2021.1914097.

[37] D. Lakens, « The Practical Alternative to the p Value Is the Correctly Used p Value », *Perspect. Psychol. Sci.*, vol. 16(3) 639–648, p. 10, févr. 2021, doi: 10.1177/1745691620958012.

[38] S. Roy, G. S. Bhunia, et P. K. Shit, « Spatial prediction of COVID-19 epidemic using ARIMA techniques in India », *Model. Earth Syst. Environ.*, vol. 7, nº 2, p. 1385-1391, juin 2021, doi: 10.1007/s40808-020-00890-y.

[39] A. Khalid et A. I. Sarwat, « Unified Univariate-Neural Network Models for Lithium-Ion Battery State-of-Charge Forecasting Using Minimized Akaike Information Criterion Algorithm », *IEEE Access*, vol. 9, p. 39154-39170, 2021, doi: 10.1109/ACCESS.2021.3061478.

[40] M. S. Yadnya, A. Mauludiyanto, et G. Hendrantoro, « Akaike Information Criteria Application to Stationary and Nonstationary Rainfalls for Wireless Communication Channel in Surabaya », p. 8, 2008.