IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 13, No. 2, June 2024, pp. 2323~2332
ISSN: 2252-8938, DOI: 10.11591/ijai.v13.i2.pp2323-2332

Smart traffic forecasting: leveraging adaptive machine learning and big data analytics for traffic flow prediction

Idriss Moumen, Jaafar Abouchabaka, Najat Rafalia
Department of Computer Science, Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco

Article Info

Article history:

Received Jun 26, 2023
Revised Sep 29, 2023
Accepted Dec 16, 2023

Keywords:

Big data analytics
Deep learning
Intelligent transportation systems
Machine learning
Neural networks
Traffic flow prediction

ABSTRACT

The issue of road traffic congestion has become increasingly apparent in modern times. With the rise of urbanization, technological advancements, and an increase in the number of vehicles on the road, almost all major cities are experiencing poor traffic environments and low road efficiency. To address this problem, re-searchers have turned to diverse data resources and focused on predicting traffic flow, a crucial issue in intelligent transportation systems (ITS) that can help al-leviate congestion. By analyzing data from correlated roads and vehicles, such as speed, density, and flow rate, it is possible to anticipate traffic congestion and patterns. This paper presents an adaptive traffic system that utilizes supervised ma-chine learning and big data analytics to predict traffic flow. The system monitors and extracts relevant traffic flow data, analyzes and processes the data, and stores it to enhance the model's accuracy and effectiveness. A simulation was conducted by the authors to showcase the proposed solution. The outcomes of the study car-ry substantial implications for transportation systems, offering valuable insights for enhancing traffic flow management.

Corresponding Author:

Idriss Moumen
Department of Computer Science, Faculty of Sciences, Ibn Tofail University
B.P 133, University Campus, Kenitra, Morocco
Email: idriss.moumen@uit.ac.ma

1. INTRODUCTION

The surge in the number of vehicles on roads, which has resulted in significant traffic congestion [1], is causing adverse environmental and economic impacts and reducing mobility [2]. To tackle these challenges, experts are using intelligent transportation systems (ITS) to improve traffic management and enhance the overall transportation experience [3], [4]. With the emergence of big data analytics and the proliferation of wireless communication technologies, various sources can gather an extensive volume of real-time transportation data. This data creates novel opportunities for traffic flow prediction [5], which is pivotal for traffic management, route optimization, and other ITS applications. Using statistical and machine learning (ML) techniques, predictive models can be created to detect patterns and make predictions about traffic flow. Recently, deep learning (DL), which is a ML method, has piqued the attention of both academic and industrial researchers. DL has been shown to be useful in a wide range of tasks, including classification, natural language processing, reducing dimensionality, object detection [6], and motion modeling. This has been demonstrated in numerous studies, such as [7]–[11]. DL algorithms use multi-layer or deep structures to find underlying properties in data from the most basic to the most complex. Revealing substantial amounts of structure within the data. Furthermore, due to their unique qualities, such as distributed storage and large parallel structure, neural networks have become the target of substantial research by numerous experts and scholars. Numerous studies have been conducted in this field, employing a variety of methods such as Kalman state space filtering models [12], support vector machine (SVM) models [13], neuro-fuzzy systems [14], autoregressive integrated moving average models [15], radial basis function neural network models [16], [17], Type-2 fuzzy logic approach [18], k-nearest neighbor (KNN) model [19], binary neural network models [20], [21], Bayesian network models [22], [23], back propagation neural network models [24], [25]. Recently, researchers have combined artificial neural network (ANN) with empirical mode decomposition and auto-regressive integrated moving average (ARIMA) to increase forecasting accuracy [26]. ARIMA is frequently contrasted with hybrid models like the long short-term memory (LSTM) [27]. However, but there have been comparisons between ARIMA and Facebook Prophet [28], as well as between ARIMA, LSTM, and Facebook Prophet [29]. Weytjens *et al.* [30] compared multi-layer LSTM networks with ARIMA and Facebook Prophet for forecasting cash flow or demand, while Abbasimehr *et al.* [31] used multi-layer LSTM networks to accomplish the same thing. Because of their ease of use and minimal data requirements, seasonal auto-regressive integrated moving average (SARIMA) models are common [32]. Over the years, the use of ML has become common among researchers to predict traffic injurie several models like elman recurrent neural network (ERNN) [33], LSTM [34]–[37], and extreme gradient boosting (XGBoost) [38] have been used and proven to increase the precision of forecasts. In this study, three tree ML models-logistics regression (LR), linear regressor, and decision tree (DT)-as well as two DL models-Facebook Prophet and LSTM-based on recurrent neural networks (RNNs)-are compared for the task of predicting traffic flow at an intersection. The goal is to use these models to modernize the traffic light system by improving traffic flow without having to completely alter the system, making its implementation more practical. The experiments show that all models are effective at forecasting vehicle flow and can be implemented in a smart traffic light system. The remaining parts of the essay are arranged as follows. The various data-analytics-based strategies for predicting traffic flow are described in section 2 using their respective methods. The experimental findings are discussed in section 3. Final observations are described in section 4.

2. MATERIALS AND METHODS

The data used in this research were obtained from different locations on the roads of England for 2020, which is a public dataset for traffic prediction derived from a variety of traffic sensors. Data on a 4 million rows and 37 columns with diverse data types such as strings, integers, and dates. This research uses five models-linear regression, LR, DT, LSTM and Facebook Prophet models. Figure 1 shows the typical architecture of the proposed model to predict the future traffic flow at an intersection.

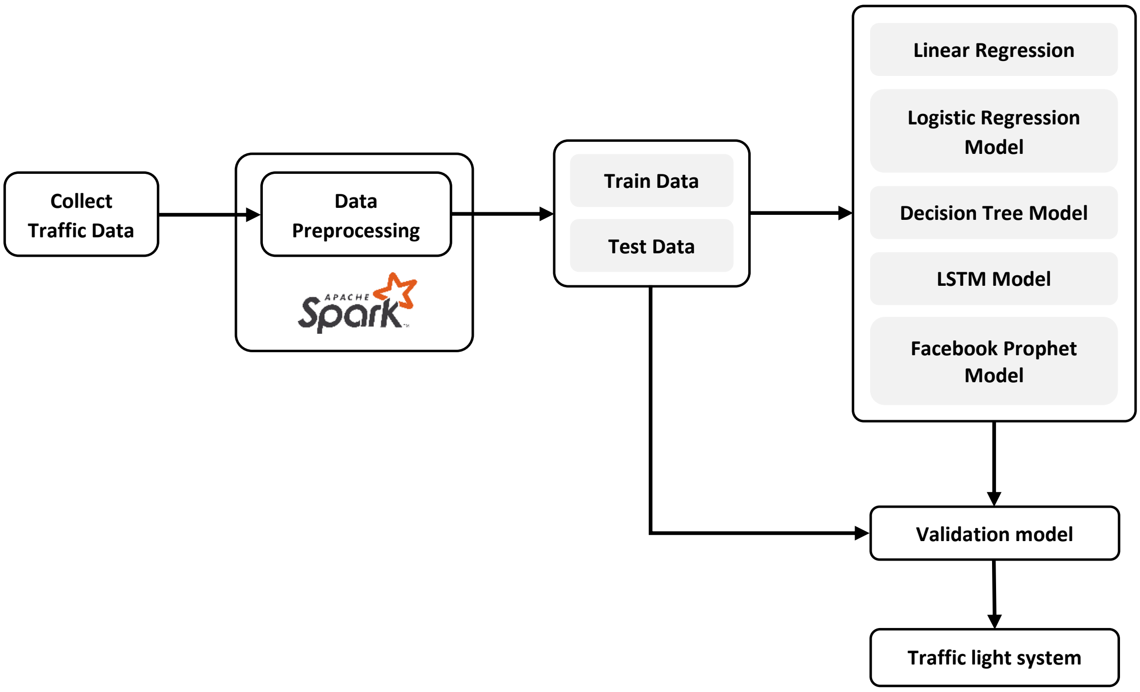

Figure 1. The flowchart presented illustrates the process of predicting traffic flow using AI-based models and experimental methods

In order to analyze preprocessed data and evaluate the effectiveness of traffic flow prediction models, three standard classification algorithms with Spark/MLlib implementations, namely LR and DT, were employed for model training and evaluation. Prior to the training and evaluation process, the data underwent preprocessing steps such as feature extraction, normalization, and principal component analysis. It is important to note that the same preprocessing techniques were applied consistently to the data before training and evaluating each algorithm.

2.1. Linear regression model

Linear regression is widely used in various applications due to its simplicity and interpretability. It is especially useful when there is a linear relationship between the input and output variables. The model's ability to provide insights into the strength and direction of the relationship between variables makes it valuable in fields like economics, finance, and social sciences. The linear regression has an equation of the (1):

$$Y = \beta_0 + \beta_1 X + \varepsilon \tag{1}$$

where Y is the response variable, X is the predictor variable, β_0 and β_1 are the regression coefficients or regression parameters, and ε is the error term. The regression coefficients β_0 and β_1 determine the intercept and slope of the regression line, respectively. The error term ε accounts for the discrepancy between the predicted data and the observed data, as it represents the unexplained variability in the response variable not captured by the predictor variable. The predicted value form of the predicted data is (2):

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X + \varepsilon \tag{2}$$

In regression analysis, the term $\hat{Y}$ represents the fitted or predicted value, while $\hat{\beta}$ represents the estimated regression coefficients. The fitted value is calculated based on the observed data used to derive the estimates of the regression coefficients $\hat{\beta}$, corresponding to one of the n observations in the dataset. On the other hand, the predicted values are generated for any arbitrary set of predictor variable values different from those present in the observed data. In essence, the fitted value is specific to the observed data points used during model training, while predicted values can be generated for any new combination of predictor variables beyond the scope of the training data.

2.2. Logistic regression model

LR is a classification technique that belongs to the group of linear regression methods. These methods are mathematically formulated as convex optimization problems. The primary objective of LR is to identify a set of weights that, when linearly combined, effectively forecast a dependent variable while minimizing the discrepancy between the predicted and actual values. In formal terms, the optimization problem entails finding the optimal vector of weights, denoted as w, which minimizes the loss function L. This problem is posed within the context of n training data feature vectors, represented as x_i, with a length of d, along with their corresponding labels, denoted as y_i.

$$min_{w \in \mathbb{R}^d} f(x) = \frac{1}{n} \sum_{i=1}^{n} L(w, x_i, y_i) \tag{3}$$

when training LR models, the employed loss function is the logistic loss function, which operates on a linear combination of weights and features (the $w^T x$ term). This loss function is specifically designed for LR and facilitates the optimization process by quantifying the discrepancy between predicted values and actual labels ($w^T x$ term).

$$L(w, x, y) = \log(1 + e^{-y w^T x}) \tag{4}$$

The trained model utilizes a logistic sigmoid function to make predictions by transforming the linear combination of features and weights. This sigmoid function is commonly used in binary classification tasks, where the output is mapped to a probability score between 0 and 1. By applying the sigmoid function, the model can convert the raw linear combination into a probability, allowing it to determine the likelihood of a binary outcome, such as whether an event will occur or not.

$$f(w; x) = \frac{1}{(1 + e^{-w^T x})} \tag{5}$$

$$class(w; x) = \begin{cases} 1, & f(w; x) > 0.5 \\ 0, & f(w; x) \le 0.5 \end{cases} \tag{6}$$

2.3. Decision trees model

Classification models, such as DTs, partition the solution space into binary classes through recursive splitting. This process involves creating a large DT with branches based on a chosen metric. In this study, entropy and Gini impurity were considered as metrics for determining the splitting criterion at each branch. Entropy aims to maximize the information gain of the split, rapidly narrowing down the predicted state choices. Formally, the split s is selected at each tree node to divide the dataset D of size N into two subsets, D_{left} and D_{right} with sizes N_{left} and N_{right} The objective is to maximize the entropy $E(x)$ relative to the number of discrete classes C where f_i represents the frequency of class i at a node.

$$E(x) = \sum_{i=1}^{C} -f_i \log f_i \tag{7}$$

$$arg_s \max\left(E(D) - \frac{N_{left}}{N} E(D_{left}, \mathrm{S}) - \frac{N_{right}}{N} E(D_{right}, \mathrm{S})\right) \tag{8}$$

The Gini impurity is a metric used in DT algorithms to measure the degree of impurity or uncertainty in a particular split. Unlike the information gain, which aims to find the most efficient split, the Gini impurity seeks to minimize the chances of misclassification after the split. It computes the impurity index $G(x)$ based on the probability of misclassifying a randomly chosen element from the data distribution at a specific node, and then the DT algorithm selects the split that maximizes this Gini impurity value. The Gini impurity $G(x)$ is computed as (9) and (10):

$$G(x) = \sum_{i=1}^{C} -f_i \log f_i \tag{9}$$

$$arg_s \max\left(G(D) - \frac{N_{left}}{N} G(D_{left}, \mathrm{S}) - \frac{N_{right}}{N} G(D_{right}, \mathrm{S})\right) \tag{10}$$

2.4. Long short-term memory model

LSTM, a prominent model within RNNs, is widely employed due to its capacity to retain information from sequential data [39]. Typically, an LSTM configuration with a hidden layer comprising 4 neurons and a single output producing 12 predicted values is utilized. Training occurs over 250 epochs, and it is imperative that the training set remains unshuffled. Figure 2 offers an illustrative depiction of the LSTM predictor's implementation.

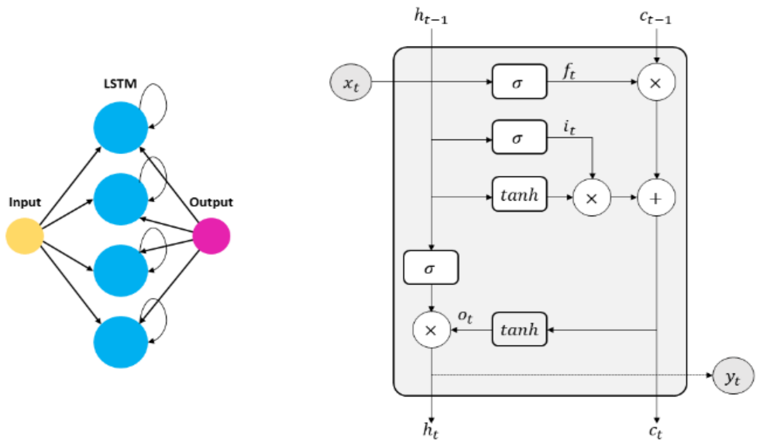

Figure 2. The LSTM cell consists of an input gate, an output gate and a forget gate

LSTM, is a type of RNN known for its ability to handle long-term dependencies in sequential data. It comprises three crucial gates: the forget gate, the input gate, and the output gate. Each gate performs a specific function in the processing of information within the network. The forget gate determines which information from the previous time step should be discarded, the input gate decides what new information to incorporate, and the output gate regulates the output state of the LSTM cell. These gates enable LSTMs to selectively retain important information, learn relevant patterns, and update their internal states, making them particularly effective in tasks involving sequential data, such as natural language processing and time series analysis. The notations are as follows: x_t is the input value of the current time step, f_t is the forget gate of the current time step, h_t and h_{t-1} is hidden states representing short-term memory for the current and previous time steps, c_t is cell states representing long-term memory for the current and previous time steps, σ is sigmoid activation function, $tanh$ is non-linear activation function allowing error learning in multiple neuron layers, i_t is input gate for the current time step, and o_t is output gate for the current time step.

The forgetting gate plays a crucial role in RNNs and LSTM networks. Its main function is to evaluate the importance of information stored in the intermediate and previous layers of the network. By using a mathematical representation, the forgetting gate determines which information should be retained for the current task and which should be discarded, thereby allowing the model to focus on the most relevant information for making accurate predictions or solving specific problems. The forgetting gate can be mathematically represented in (11):

$$f_t = \sigma(x_t \times W_f + h_{t-1} \times W_f) \tag{11}$$

– Input gate: following the forgetting gate, the input gate updates and integrates data into the memory cell using an activation function. The specific formula for the input gate is as (12):

$$i_t = \sigma(x_t \times W_i + h_{t-1} \times W_i) \tag{12}$$

– Output gate: the output gate governs the model's output by incorporating the weight of the control state c_t with the current LSTM hidden layer. The initial output is obtained through an activation function and subsequently normalized using the tanh function. The expression for the output gate is as (13) and (14):

$$o_t = \sigma(x_t \times W_o + h_{t-1} \times W_o) \tag{13}$$

$$h_t = o_t \times tanh(c_t) \tag{14}$$

2.5. Facebook Prophet model

The Prophet model, introduced by Facebook Inc. in 2017 [28], is an additive model specifically designed for time series prediction. According to Google's official presentation [40], it demonstrates superior performance when applied to time series data with pronounced seasonal effects and an ample historical data spanning multiple seasons. Prophet exhibits resilience in handling missing data, trend shifts, and outliers [41]. Since its release, the Prophet model has gained significant popularity in the field of time series analysis. It decomposes the time series into three key components: the seasonal term S_t, the trend term T_t, and the residual term R_t are as (15):

$$y_t = S_t + T_t + R_t \tag{15}$$

The Prophet model goes beyond basic time series forecasting by incorporating the influence of holidays, denoted as $h(t)$, into its predictions. This integration allows the model to account for the significant variations in data patterns that often occur during holidays. By considering the impact of holidays on the time series, the Prophet model becomes more adaptable and accurate in capturing real-world scenarios, making it a valuable tool for forecasting in diverse industries and applications.

$$y_t = g(t) + s(t) + h(t) + \varepsilon_t \tag{16}$$

The model's robustness and capacity to handle missing data and outliers make it stand out in the field of data analysis. Its ability to fit a diverse range of data with reasonable accuracy further cements its popularity among data analysts, especially when dealing with time series prediction tasks. With its versatility and reliable performance, this model has become a preferred choice for tackling complex real-world datasets, enabling analysts to make more informed decisions and predictions.

2.6. Model evaluation

After completing the training phase, it is crucial to evaluate the prediction model's accuracy using testing data. In comparing forecasting methods that share the same unit, the widely utilized metric is the root mean square error (RMSE). The RMSE is calculated using a specific equation, which quantifies the differences between the predicted values and the actual values in the testing dataset. This evaluation metric provides a measure of the model's predictive performance and allows for meaningful comparisons between different forecasting approaches.

$$\text{RMSE}(y, \hat{y}) = \sqrt{\frac{1}{n}\sum_{i=0}^{n-1}(y_i - \hat{y}_i)^2} \tag{17}$$

Where n is number of samples, y is observed traffic flow, and $\hat{y}$ is predicted traffic flow.

2.7. Data exploration

This analysis aids in identifying attribute pairs with high correlation, offering insights into the key attributes for model construction. Figure 3 displays the correlation matrix, which showcases strong correlations among attributes such as start_junction_road_name, end_junction_road_name, day, month, year, road_name, red-time, and green-time. The correlation matrix allows us to discern the relationships between these attributes and their potential impact on traffic patterns, enabling us to make informed decisions during the model development process.

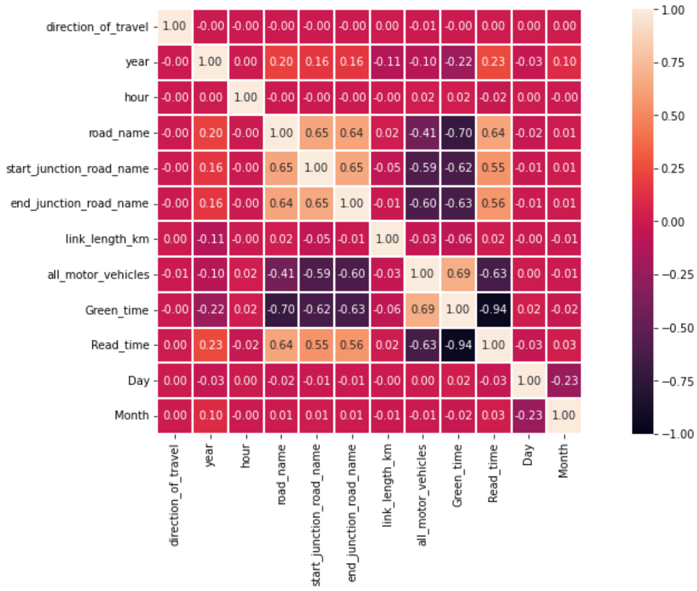

Figure 3. Correlation matrix as heat map shows good correclation between day, month, year, roadname, start junction road name, end junction road name, red time, green time

2.8. Data preprocessing

It is crucial to identify the types of data we'll be working with. In some cases, data might contain irrelevant or absent components. To address this, we perform data cleaning, which encompasses handling missing or noisy data, among other tasks. In this paper, we employed two data cleaning techniques: i) the removal of outliers and redundant values (as we discovered several of these in the dataset), and ii) the imputation of missing values (various methods exist for this purpose, but we opted to manually fill in missing values by using the attribute's mean or the most probable value). The primary goal of this approach is to transform the data into formats that are appropriate for the data exploration process. This operation involves the following techniques: Normalization: scaling data values within a specified range such as (0.0 to 1.0 or -1.0 to 1.0). This is done for categorical columns to transform them into numerical values for processing in ML algorithms; Attribute selection: in this strategy, new attributes are constructed from a determined set of attributes to simplify the data exploration process. New columns are created based on a combination of other attributes to obtain transformations such as normalization, standardization, scaling, and pivoting. Binning can also be applied (based on the number of values, and treating missing values as a separate group). Data replacement techniques can also be used, such as splitting, merging, and slicing.

3. RESULTS AND DISCUSSION

In order to maintain a systematic approach to the development process, the findings are elucidated in discrete sub-chapters. ML models, specifically LR, linear regression, and DT, were utilized in this study, in addition to time series models including LSTM and Facebook Prophet. PySpark was employed as the framework for implementing these models.

3.1. Machine learning models

Table 1 presents the performance metrics of various ML models. Three different algorithms, namely LR, linear regression, and DT, were evaluated using PySpark. For LR, the RMSE was 214.40, representing the average difference between predicted and actual values. The R2 value, indicating the proportion of variance explained by the independent variables, was 0.907. In contrast, linear regression achieved an RMSE of 140.126, indicating a lower average difference compared to LR. The R2 value was higher at 0.923, indicating a better fit to the data. The DT model yielded a higher RMSE of 306.145, indicating a relatively larger average difference compared to logistic and linear regression. The R2 value for DT was 0.877, slightly lower than the other models. Overall, linear regression exhibited superior performance with the lowest RMSE and highest R2 value among the evaluated models, highlighting it is predictive power in this context.

Table 1. Comparison of ML models performance metrics

Model	RMSE	R2
Logistic regression	214.40	0.907
Linear regression	140.126	0.923
Decison tree	306.145	0.877

3.2. Time series models

In this study, various time series models were employed to analyze and forecast events based on historical data. Common types of time series models include ARIMA, smoothing-based models, and moving average models. It is essential to consider that different models can yield different results when applied to the same dataset, emphasizing the importance of selecting the most suitable model for a specific time series analysis. In our study, we utilized LSTM and Facebook Prophet as specialized time series models for the purpose of analysis and forecasting. The results obtained from applying the LSTM model on our dataset demonstrated its effectiveness, as the predicted values of the number of vehicles closely resembled the actual values. Figure 4 depicts a plot that compares the actual data with the predictions made by the LSTM model on the testing dataset, providing visual evidence of the model's accuracy.

The LSTM neural network was implemented in Python, and prior to training the model, the data was normalized to enhance efficiency. Figure 5 provides a visual representation of the loss function performance during the training phase. The chart demonstrates a consistent decrease in the loss values for the training set, indicating that the LSTM model was effectively trained without overfitting. Unlike other models, the LSTM predictor takes into account recent values, reducing the influence of seasonality and incorporating the current trend. Additionally, the Prophet model excels in modeling as an additive system and displays proficiency in identifying and showcasing seasonality patterns.

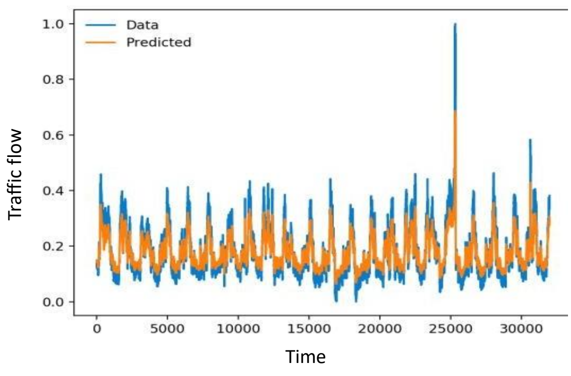

Figure 4. A plot between actual and prediction data on testing data for number of vehicles with LSTM model

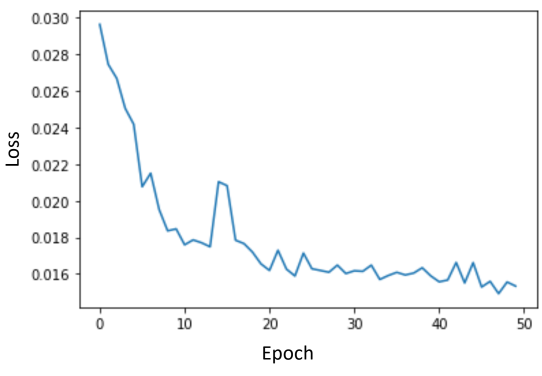

Figure 5. Loss function performance during the training phase

Figure 6 displays the time series graph generated using the Facebook Prophet model. The graph illustrates a clear upward trend in the data, indicating an overall increase in values over time. Additionally, there is a possibility of a slight curvature in the data, as the rate of increase appears to be accelerating. In such cases, the quadratic model becomes a suitable choice for capturing the underlying pattern. It is worth noting that the time series data does not exhibit a distinct upward or downward trend in general. The higher average consumption observed in previous years can be attributed to the lack of recent data, during which road occupancy was high. Therefore, when comparing year-by-year data, the road occupancy should remain relatively stable.

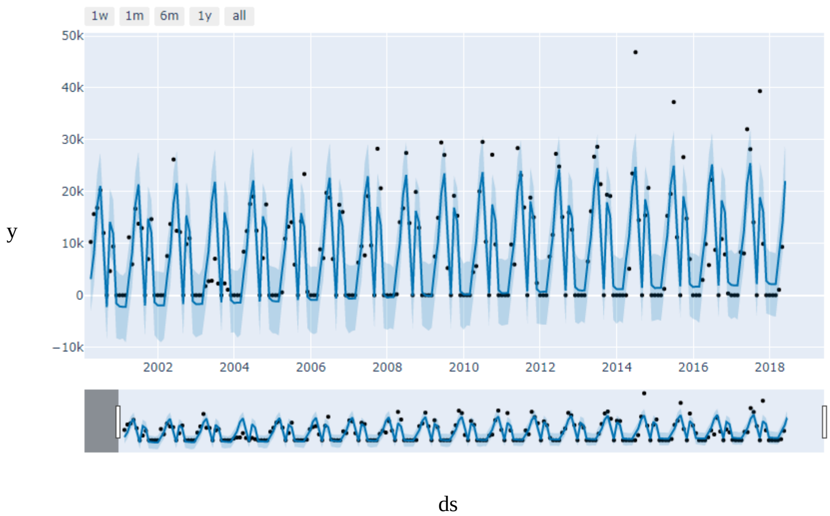

Figure 6. The time series graph realized with the Facebook prophet model shows a clear upward trend

4. CONCLUSION

The escalating number of vehicles on roads has led to significant traffic congestion, causing detrimental environmental and economic consequences while hampering mobility. To address these challenges, experts have turned to ITS as a means to enhance traffic management and improve the overall transportation experience. The advent of big data analytics and the proliferation of wireless communication technologies have facilitated the collection of extensive real-time transportation data, opening up new opportunities for traffic flow prediction. Statistical and ML techniques have been leveraged to develop predictive models capable of detecting patterns and making accurate predictions about traffic flow. DL, a powerful ML method, has garnered significant attention from both academia and industry for it is remarkable capabilities in various tasks, including object detection, motion modeling, and natural language processing. Researchers have explored diverse approaches, such as Kalman state space filtering models, SVM models, neuro-fuzzy systems, and neural network models, to predict traffic flow. Recently, combining ANN with empirical mode decomposition and ARIMA has shown promising results in increasing forecasting accuracy. Comparisons have been made between different models, including ARIMA, LSTM, and Facebook Prophet, highlighting the strengths and limitations of each. In this study, five models-LR, linear regressor, DT, Facebook Prophet, and LSTM-were evaluated for the task of predicting traffic flow at an intersection, aiming to enhance traffic light systems without significant system overhauls. The experimental results demonstrate the effectiveness of all models in forecasting vehicle flow and their potential implementation in smart traffic light systems.

REFERENCES

[1] K. Saito *et al.*, "A regulatory circuit for piwi by the large Maf gene traffic jam in Drosophila," *Nature*, vol. 461, no. 7268, pp. 1296–1299, Oct. 2009, doi: 10.1038/nature08501.
[2] P. Anttila, J.-P. Tuovinen, and J. V. Niemi, "Primary NO2 emissions and their role in the development of NO2 concentrations in a traffic environment," *Atmospheric Environment*, vol. 45, no. 4, pp. 986–992, Feb. 2011, doi: 10.1016/j.atmosenv.2010.10.050.
[3] R. Sennett, *The uses of disorder: Personal identity and city life*. Brooklyn, New York: Verso, 2021.
[4] F. Chuang-Lin and W. De-Li, "Comprehensive measures and improvement of Chinese urbanization development quality," *Geographical Research*, vol. 30, no. 11, pp. 1931–1946, 2011.
[5] I. Moumen, J. Abouchabaka, and N. Rafalia, "Adaptive traffic lights based on traffic flow prediction using machine learning

models," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 5, pp. 5813–5823, Oct. 2023, doi: 10.11591/ijece.v13i5.pp5813-5823.

[6] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, "A review on deep learning techniques applied to semantic segmentation," *Computing Research Repository arXiv preprint*, pp. 1-23, 2017.

[7] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006, doi: 10.1126/science.1127647.

[8] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learningA unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 160-167, 2008.

[9] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, "Multi-digit number recognition from street view imagery using deep convolutional neural networks," *International Conference on Learning Representations (ICLR)*, pp. 1-12, 2014.

[10] B. Huval, A. Coates, and A. Ng, "Deep learning for class-generic object detection," *arXiv- Computer Science*, pp. 1-3, 2013.

[11] H.-C. Shin, M. R. Orton, D. J. Collins, S. J. Doran, and M. O. Leach, "Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1930–1943, Aug. 2013, doi: 10.1109/TPAMI.2012.277.

[12] B. Ait-El-Fquih and I. Hoteit, "Fast Kalman-like filtering for large-dimensional linear and Gaussian state-space models," *IEEE Transactions on Signal Processing*, vol. 63, no. 21, pp. 5853–5867, Nov. 2015, doi: 10.1109/TSP.2015.2468674.

[13] Y. Zhang and Y. Liu, "Data imputation using least squares support vector machines in urban arterial streets," *IEEE Signal Processing Letters*, vol. 16, no. 5, pp. 414–417, May 2009, doi: 10.1109/LSP.2009.2016451.

[14] J. Perez, A. Gajate, V. Milanes, E. Onieva, and M. Santos, "Design and implementation of a neuro-fuzzy system for longitudinal control of autonomous vehicles," in *International Conference on Fuzzy Systems*, Jul. 2010, pp. 1–6, doi: 10.1109/FUZZY.2010.5584208.

[15] A. Guin, "Travel time prediction using a seasonal autoregressive integrated moving average time series model," in *2006 IEEE Intelligent Transportation Systems Conference*, 2006, pp. 493–498, doi: 10.1109/ITSC.2006.1706789.

[16] H. Gao, J. Zhao, and L. Jia, "Short-term traffic flow forecasting model of Elman neural network based on dissimilation particle swarm optimization," in *2008 IEEE International Conference on Networking, Sensing and Control*, Apr. 2008, pp. 1305–1309, doi: 10.1109/ICNSC.2008.4525419.

[17] L. Li, W.-H. Lin, and H. Liu, "Type-2 fuzzy logic approach for short-term traffic forecasting," *IEE Proceedings-Intelligent Transport Systems*, vol. 153, no. 1, pp. 33–40, 2006, doi: 10.1049/ip-its:20055009.

[18] A. X. Wang, S. S. Chukova, and B. P. Nguyen, "Ensemble k-nearest neighbors based on centroid displacement," *Information Sciences*, vol. 629, pp. 313–323, Jun. 2023, doi: 10.1016/j.ins.2023.02.004.

[19] X. Xu, X. Jin, D. Xiao, C. Ma, and S. C. Wong, "A hybrid autoregressive fractionally integrated moving average and nonlinear autoregressive neural network model for short-term traffic flow prediction," *Journal of Intelligent Transportation Systems*, vol. 27, no. 1, pp. 1–18, Jan. 2023, doi: 10.1080/15472450.2021.1977639.

[20] L. Kerkelä, K. Seunarine, R. N. Henriques, J. D. Clayden, and C. A. Clark, "Improved reproducibility of diffusion kurtosis imaging using regularized non-linear optimization informed by artificial neural networks,"*arXiv-Physics*, pp. 1-16, 2022.

[21] Y. Li, W. Zhao, and H. Fan, "A spatio-temporal graph neural network approach for traffic flow prediction," *Mathematics*, vol. 10, no. 10, pp. 1-14, May 2022, doi: 10.3390/math10101754.

[22] Y. Yang, S. Rasouli, and F. Liao, "Effects of life events and attitudes on vehicle transactions: A dynamic Bayesian network approach," *Transportation Research Part C: Emerging Technologies*, vol. 147, pp. 1-27, Feb. 2023, doi: 10.1016/j.trc.2022.103988.

[23] M. Gui, A. Pahwa, and S. Das, "Bayesian network model with Monte Carlo simulations for analysis of animal-related outages in overhead distribution systems," *IEEE Transactions on Power Systems*, vol. 26, no. 3, pp. 1618–1624, Aug. 2011, doi: 10.1109/TPWRS.2010.2101619.

[24] N. Brahimi, H. Zhang, L. Dai, and J. Zhang, "Modelling on car-sharing serial prediction based on machine learning and deep learning," *Complexity*, vol. 2022, pp. 1–20, Jan. 2022, doi: 10.1155/2022/8843000.

[25] C. Li, Y. Xie, H. Zhang, and X. Yan, "Dynamic division about traffic control sub-area based on back propagation neural network," in *2010 Second International Conference on Intelligent Human-Machine Systems and Cybernetics*, Aug. 2010, pp. 22–25, doi: 10.1109/IHMSC.2010.104.

[26] Ü. Ç. Büyükşahin and Ş. Ertekin, "Improving forecasting accuracy of time series data using a new ARIMA-ANN hybrid method and empirical mode decomposition," *Neurocomputing*, vol. 361, pp. 151–163, Oct. 2019, doi: 10.1016/j.neucom.2019.05.099.

[27] W. Chen, H. Xu, L. Jia, and Y. Gao, "Machine learning model for Bitcoin exchange rate prediction using economic and technology determinants," *International Journal of Forecasting*, vol. 37, no. 1, pp. 28–43, Jan. 2021, doi: 10.1016/j.ijforecast.2020.02.008.

[28] S. J. Taylor and B. Letham, "Forecasting at scale," *Peerj Preprints*, vol. 5, pp. 1–25, 2017, doi: 10.1080/00031305.2017.1380080.

[29] N. K. Chikkakrishna, C. Hardik, K. Deepika, and N. Sparsha, "Short-term traffic prediction using sarima and FbPROPHET," in *2019 IEEE 16th India Council International Conference (INDICON)*, Dec. 2019, pp. 1–4, doi: 10.1109/INDICON47234.2019.9028937.

[30] H. Weytjens, E. Lohmann, and M. Kleinsteuber, "Cash flow prediction: MLP and LSTM compared to ARIMA and Prophet," *Electronic Commerce Research*, vol. 21, no. 2, pp. 371–391, Jun. 2021, doi: 10.1007/s10660-019-09362-7.

[31] H. Abbasimehr, M. Shabani, and M. Yousefi, "An optimized model using LSTM network for demand forecasting," *Computers and Industrial Engineering*, vol. 143, May 2020, doi: 10.1016/j.cie.2020.106435.

[32] S. Siami-Namini, N. Tavakoli, and A. Siami Namin, "A comparison of ARIMA and LSTM in forecasting time series," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec. 2018, pp. 1394–1401, doi: 10.1109/ICMLA.2018.00227.

[33] K. Mehmood, H. T. Ul Hassan, A. Raza, A. Altalbe, and H. Farooq, "Optimal power generation in energy-deficient scenarios using bagging ensembles," *IEEE Access*, vol. 7, pp. 155917–155929, 2019, doi: 10.1109/ACCESS.2019.2946640.

[34] C. Fan, K. Matkovic, and H. Hauser, "Sketch-based fast and accurate querying of time series using parameter-sharing LSTM network," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 12, pp. 4495–4506, Dec. 2021, doi: 10.1109/TVCG.2020.3002950.

[35] M. Qiao, S. Yan, X. Tang, and C. Xu, "Deep convolutional and LSTM recurrent neural networks for rolling bearing fault diagnosis under strong noises and variable loads," *IEEE Access*, vol. 8, pp. 66257–66269, 2020, doi: 10.1109/ACCESS.2020.2985617.

[36] Y. Cui, H. Xu, J. Wu, Y. Sun, and J. Zhao, "Automatic vehicle tracking with roadside LiDAR data for the connected-vehicles system," *IEEE Intelligent Systems*, vol. 34, no. 3, pp. 44–51, May 2019, doi: 10.1109/MIS.2019.2918115.

[37] M. F. Tahir, C. Haoyong, K. Mehmood, N. A. Larik, A. Khan, and M. S. Javed, "Short term load forecasting using bootstrap aggregating based ensemble artificial neural network," *Recent Advances in Electrical and Electronic Engineering (Formerly Recent Patents on Electrical and Electronic Engineering)*, vol. 13, no. 7, pp. 980–992, Nov. 2020, doi: 10.2174/2213111607666191111095329.

[38] J. Luo, Z. Zhang, Y. Fu, and F. Rao, "Time series prediction of COVID-19 transmission in America using LSTM and XGBoost algorithms," *Results in Physics*, vol. 27, pp. 1-9, Aug. 2021, doi: 10.1016/j.rinp.2021.104462.
[39] R. Yu *et al.*, "LSTM-EFG for wind power forecasting based on sequential correlation features," *Future Generation Computer Systems*, vol. 93, pp. 33–42, Apr. 2019, doi: 10.1016/j.future.2018.09.054.
[40] C. Xie *et al.*, "Trend analysis and forecast of daily reported incidence of hand, foot and mouth disease in Hubei, China by Prophet model," *Scientific Reports*, vol. 11, no. 1, pp. 1-8, Jan. 2021, doi: 10.1038/s41598-021-81100-2.
[41] B. Rostami-Tabar and J. F. Rendon-Sanchez, "Forecasting COVID-19 daily cases using phone call data," *Applied Soft Computing*, vol. 100, pp. 1-11, Mar. 2021, doi: 10.1016/j.asoc.2020.106932.

BIOGRAPHIES OF AUTHORS

Idriss Moumen born in 1990 in Kenitra. He received his Master's degree in computer science, Computer Sciences Researches from Ibn Tofail University, Kenitra, Morocco. He is a Ph.D. student in Computer Research Laboratory (LaRI) at Ibn Tofail. His research interests include big data, data mining, ML, DL, and distributed computing. He can be contacted at email: idriss.moumen@uit.ac.ma.

Jaafar Abouchabaka was born in Guersif, Morocco, 1968. He has obtained two doctorates in Computer Sciences applied to mathematics from Mohammed V University, Rabat, Morocco. Currently, he is a professor at Department of computer Sciences, Ibn Tofail University, Kenitra, Morocco. His research interests are in concurrent and parallel programming, distributed systems, multi agent systems, genetics algorithms, big data, and cloud computing. She can be contacted at email: jaafar.abouchabaka@uit.ac.ma.

Najat Rafalia was born in Kenitra, Morocco, 1968. She has obtained three doctorates in Computer Sciences from Mohammed V University, Rabat, Morocco by collaboration with ENSEEIHT, Toulouse, France, and Ibn Tofail University, Kenitra, Morocco. Currently, she is a professor at Department of Computer Sciences, Ibn Tofail University, Kenitra, Morocco. Her research interests are in distributed systems, multi-agent systems, concurrent and parallel programming, communication, security, big data, and cloud computing. He can be contacted at email: arafalia@yahoo.com.