International Journal of Electrical and Computer Engineering (IJECE)
Vol. 13, No. 5, October 2023, pp. 5813~5823
ISSN: 2088-8708, DOI: 10.11591/ijece.v13i5.pp5813-5823

Adaptive traffic lights based on traffic flow prediction using machine learning models

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

Article Info

Article history:

Received Jan 9, 2023
Revised Mar 20, 2023
Accepted Apr 7, 2023

Keywords:

Adaptive traffic light system
Intelligent transport systems
Machine learning
Traffic prediction
Traffic simulation

ABSTRACT

Traffic congestion prediction is one of the essential components of intelligent transport systems (ITS). This is due to the rapid growth of population and, consequently, the high number of vehicles in cities. Nowadays, the problem of traffic congestion attracts more and more attention from researchers in the field of ITS. Traffic congestion can be predicted in advance by analyzing traffic flow data. In this article, we used machine learning algorithms such as linear regression, random forest regressor, decision tree regressor, gradient boosting regressor, and K-neighbor regressor to predict traffic flow and reduce traffic congestion at intersections. We used the public roads dataset from the UK national road traffic to test our models. All machine learning algorithms obtained good performance metrics, indicating that they are valid for implementation in smart traffic light systems. Next, we implemented an adaptive traffic light system based on a random forest regressor model, which adjusts the timing of green and red lights depending on the road width, traffic density, types of vehicles, and expected traffic. Simulations of the proposed system show a 30.8% reduction in traffic congestion, thus justifying its effectiveness and the interest of deploying it to regulate the signaling problem in intersections.

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 intelligent transportation system is an essential part of the smart city [1]–[3]. A thorough understanding of how these systems work and the development of effective traffic management and control strategies are required to optimize their management. Traffic prediction is crucial in traffic systems. An accurate and efficient traffic flow prediction system is needed to achieve intelligent transport systems (ITS) [4], [5]. The traffic flow prediction system is used to provide accurate road status information [6]. However, traffic congestion is a growing global problem as the population increases. Therefore, each traffic intersection is likely to gather a lot of vehicles, and the peak period will be more congested than usual. Consequently, to solve this problem, numerous research studies have been carried out to explore and manage traffic flow to reduce traffic congestion. Traffic flow data were collected from the Tokyo Expressway. These data were used as input to the predictive model [7]. In [8], the model predicted the value of traffic speed on each road section at the next time stamp using data collected from Beijing. By including a multi-task layer at the conclusion of a deep belief network (DBN) [9]. Concurrently predicted traffic speed and flow. While in [10], to forecast traffic in the upcoming six timestamps simultaneously, the authors used both the iteration approach and the auto-correlation coefficient method. Regression models are easy to implement and suited for traffic prediction tasks on a simple traffic network. According to [11], the mathematical model between inputs and outputs and associated

parameters are predetermined, and the relationship between each parameter and input data is relatively certain. The experimental results also show that the linear regression model is more efficient and can provide satisfactory prediction results.

There are many types of classification methods for machine learning models based on different perspectives [12]–[17]. In addition, the studies [18], [19] demonstrate the robust performance of linear regression models. The authors provide a forecasting method that combines linear regression with time correlation coefficients. Linear regression deals with the spatio-temporal relationship of road networks and can also use this information to make predictions. Hobeika and Kim [20] provided prediction models, each model combined with different traffic variables such as upstream, current traffic and historical average. Kwon *et al.* [21] combined linear regression with stepwise-variable-selection method and tree-based method. To avoid the intersection problem, the authors used cross-validation (CV). According to [22], the k-nearest neighbor (KNN) model is a data-based non-parametric regression method. KNN model finds the k-nearest neighbors that match the current variable value and use those K data to predict the next period's value. Furthermore, in [23], four main challenges were pointed out: i) how to define an appropriate state vector, ii) how to define a suitable distance metric, iii) how to generate forecast, and iv) how to manage the potential neighbor database. In [24], KNN model was used for traffic flow prediction in different prediction intervals (15, 30, 45, 60 min). To improve the K nearest neighbors selection process, Zheng and Su [25] chose correlation coefficient distance as the distance metric to select the more related neighbors. The authors then introduced a linearly sewing principal component method (LSPC), which steers the final prediction into a minimization problem. A rather different approach was taken in [26], in which MapReduce-based KNN was introduced for traffic flow prediction. The authors also employed distance weighted voting to mitigate the impact of the number of K.

Various techniques have been evaluated in the literature. In [27], [28], chose the support vector machines (SVM) model as the kernel of the online traffic prediction system. SVM need less computational requirements. However, SVM has difficulties dealing with a big traffic network problem. Furthermore, when using grid search, the cost of finding appropriate hyper-parameters is relatively high when compared to KNN or regression models [28]. However, methods such as continuous ant colony optimization (CACO) [29] and genetic particle swarm optimization (GPSO) [30] are used to quickly determine the hyper-parameters values. According to [23], [31], when compared to the traditional linear regression model, the non-parametric method can better capture the non-linear feature of the traffic pattern, which greatly increases the algorithm's precision. However, the non-parametric method also comes with drawbacks [32]. This model imposes high demands on the quantity and quality of historical data, and training costs for nonparametric models can be high compared to other types of methods. Recurrent neural networks (RNN) were introduced to solve this problem. RNNs can better describe the impact of a series of traffic records [33]–[35]. RNNs also have drawbacks. If the input series is too long during the optimization process, it can lead to gradient explosion or gradient vanishing problems. To avoid this problem, Hochreiter and Schmidhuber [36] proposed long-short term memory (LSTM) model to solve the long-term dependency. as highlighted by [37], LSTM is limited when the input time series is very long. Fu *et al.* [38] compared gated recurrent unit (GRU) and LSTM by using the traffic data collected by 30 random sensors from the performance management system (PeMS) dataset. A strategy for managing traffic is to use traffic lights [39] with sporadic fixed times for each road, but this cannot be considered a good solution because congestion will increase on specific roads with high vehicle traffic.

Furthermore, the use of cameras is complemented by modern technologies such as computer graphics, automated video analysis for traffic monitoring, state-of-the-art cameras and high-end computing power [40]. Delica *et al.* [41], a simulation program for an intelligent traffic light system was created using the virtual instrumentation laboratory environment (LabVIEW). It intends to measure traffic density by examining the number of vehicles in each lane. Kareem and Hoomod [42] presents an integrated three-part module for intelligent traffic signal systems. This paper presents five machine learning models: linear regression, random forest regressor, decision tree regressor, gradient boosting regressor, K neighbors regressor; compared in the task of predicting traffic flow at intersections, with the purpose of applying them to an adaptive traffic light system, this method allows for a better traffic flow without completely changing the traffic light system structure. Experiments showed that all models have good vehicular flow prediction capabilities and can be used effectively to develop our traffic management system. The rest of this paper is organized as follows: section 2 explores several machine learning algorithms that has being employed in this work; section 3 discusses the experimentation and evaluation; finally, section 4 presents the conclusion and perspectives.

2. METHOD AND IMPLEMENTATION

The goal of this study is to reduce traffic congestion. The measure for traffic congestion is based on the density of vehicles on the roadways being monitored. Density is a commonly used measure of traffic

congestion, and it refers to the number of vehicles present on a particular stretch of road at a given time. The proposed architecture is as follows: We first collect traffic data from cameras and sensors, then deploy the cross-industry standard process for data mining (CRISP-DM) model to process the collected data and apply machine learning algorithms. After implementing ML algorithms, we apply the grid search method to find the best hyper-parameters. Next, a density reduction function is then applied based on the change of the green-light and red-light duration. Finally, we feed the obtained results to the traffic light controller. This model is better illustrated in Figure 1.

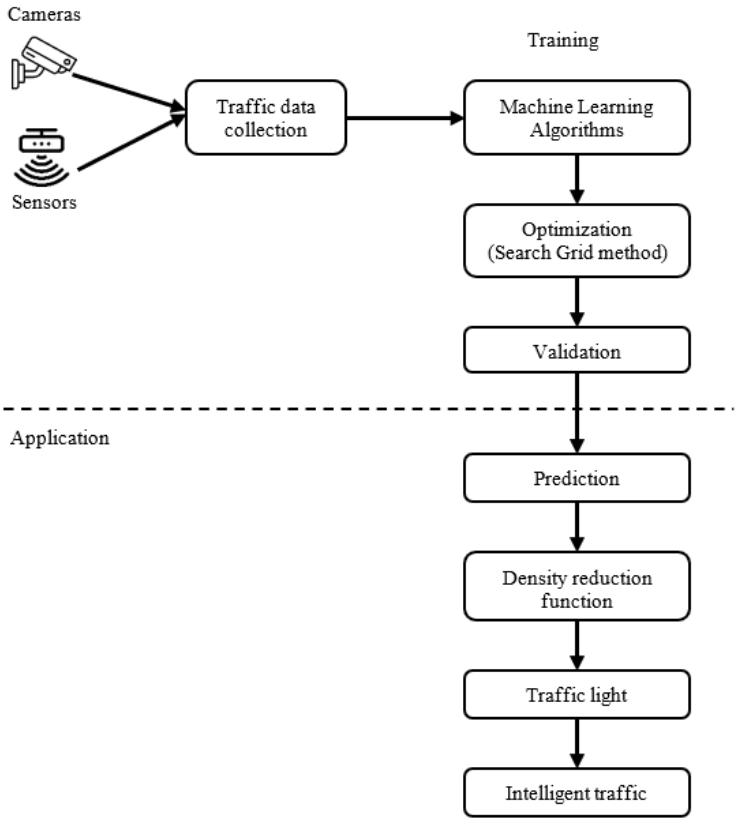

Figure 1. The proposed adaptive traffic light system

In this paper, the UK National Road traffic estimation statistics publication's road dataset was used to train, validate, and test the proposed system. This is a public dataset for traffic prediction derived from a variety of traffic sensors. It is important to note that only a few public transport data are available.

Beginning our analysis with some form of exploratory data analysis is consistently a wise decision. This way, we can determine dataset properties such as data format, number of samples, and data types. The dataset contains a total of 495,562 data points with 32 attributes. A sample data is shown in Figure 2. The dataset's variables were categorized as either categorical or numerical depending on their characteristics. Handling a large number of features may have an impact on the model's performance and increase the risk of overfitting, as training time grows exponentially with the number of features.

Using a measure of centrality is the easiest and quickest approach to imputing missing values [43]. Such measures represent the most common value in a variable's distribution and are therefore a logical option for this method. There are multiple measures of centrality available, such as the mean and median. The most appropriate measure depends on the distribution of the variable. Figure 3 shows that the data follows a normal distribution, where all data points are closely gathered around the mean. This measure is the optimal choice to impute the missing values.

The second step is to explore the relationships between variables. This analysis enables us to identify the most correlated pairs, pairs that are highly correlated represent the same variance of the data, this allows us to further investigate the pairs to better understand which attribute is most important for constructing our models. The correlation matrix in Figure 4 shows a strong correlation between *start_iunction_Raod_name, end_junction_raod-name, day, month, year, road_name, red-time,* and *green-time attributes*.

	count_point_id	direction_of_travel	year	count_date	hour	region_id	region_name	local_authority_id	local_authority_name	road_name	...	buses_and_coach
0	940897	E	2003	2003-03-18	7	7	East of England	97	Cambridgeshire	B1443	...	
1	940897	E	2003	2003-03-18	8	7	East of England	97	Cambridgeshire	B1443	...	
2	940897	E	2003	2003-03-18	9	7	East of England	97	Cambridgeshire	B1443	...	
3	940897	E	2003	2003-03-18	10	7	East of England	97	Cambridgeshire	B1443	...	
4	940897	E	2003	2003-03-18	11	7	East of England	97	Cambridgeshire	B1443	...	
...	...	...	...	...	...	...	...	...	...	...	...	
495559	6672	E	2002	2002-10-02	14	7	East of England	123	Essex	A133	...	
495560	6672	E	2002	2002-10-02	15	7	East of England	123	Essex	A133	...	
495561	6672	E	2002	2002-10-02	16	7	East of England	123	Essex	A133	...	
495562	6672	E	2002	2002-10-02	17	7	East of England	123	Essex	A133	...	
495563	6672	E	2002	2002-10-02	18	7	East of England	123	Essex	A133	...	

495562 rows × 32 columns

Figure 2. A sample of the used dataset containing quantitative and qualitative data

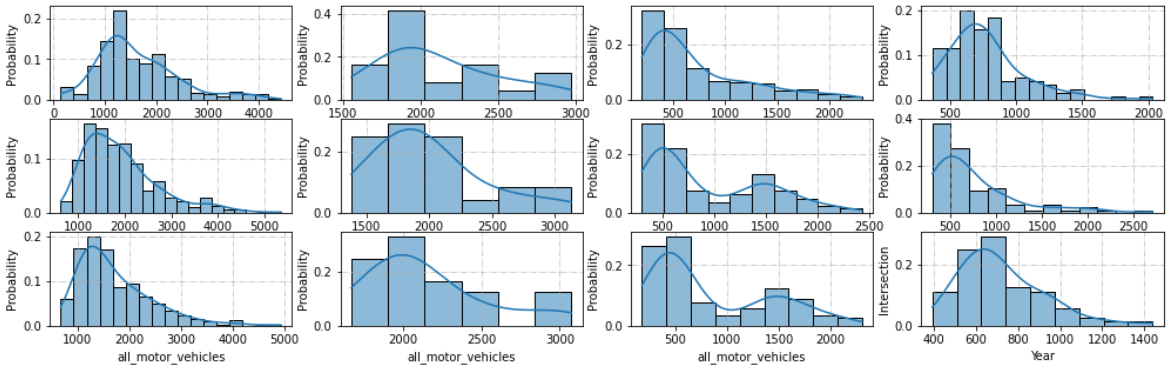

Figure 3. Distribution of vehicles by year and by road across all sensing segments

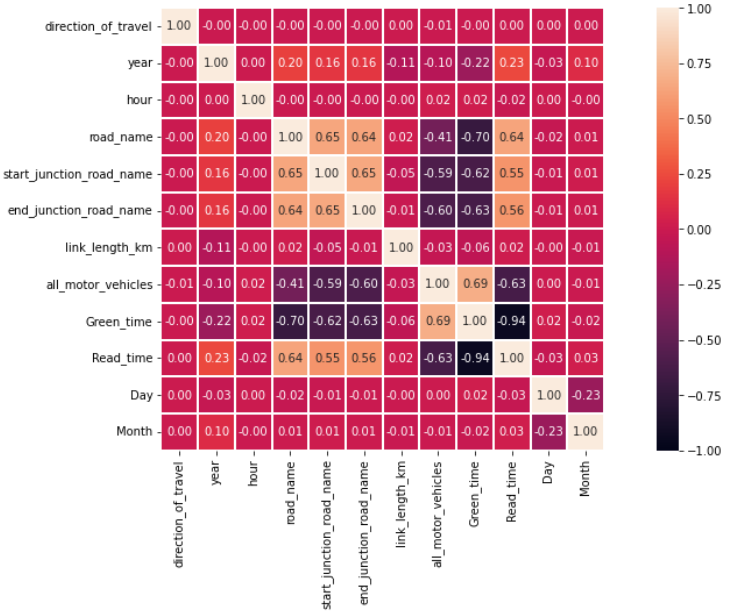

Figure 4. Correlation matrix as Heat Map shows good correlation between *day*, *month*, *year*, *raodName*, *startJunctionRaodName*, *endJunctionRaodName*, *redTime*, and *greenTime*

2.1. Data cleaning

A common practice during this task is to correct or delete erroneous values. The problem of dealing with missing data is an important issue, as this data cannot be ignored in a machine learning analysis. This issue can be solved by imputation techniques which consist in producing an "artificial value" to replace the missing value to generate approximately unbiased estimates. For this we chose the method of imputation by average which consists in replacing the null values by the average of the whole column. And also, the method more often for categorical fields. We have applied two functions that go through all the quantitative, qualitative data and replace them using the impute function in the sklearn library.

2.2. Encoding of categorical characteristics

This is an important preprocessing step for structured data sets in supervised learning. Most machine learning algorithms only utilize numerical data. Our dataset contains four categorical variables that must be encoded: *direction_of_travel*, *start_junction_road_name*, *road_name* and *end_junction_road_name*. This information is detailed about roads that are of a qualitative nature, as a result, we must encode the qualitative characteristics in a representation in order to convert the machine-readable form. This is accomplished using the LabelEncoder from the sklearn library.

2.3. Machine learning methods

Five regression models were implemented in Python using scikit-learn library [44]: linear regression, random forest regression, decision tree regression, gradient boosting regression, and K-neighborhood regression. All of these models have default parameters. Additionally, most machine learning algorithms have a large number of parameters that can be tuned to control the modeling process. These measures are called hyper-parameters and must be adjusted so that the model can obtain optimum results. Grid search method is used, which is simply an exhaustive search in a manually specified subset of the hyperparameter space of a learning algorithm, for ease of understanding a list of values is established for each hyperparameter each hyperparameter and then the one that gives the most accuracy is chosen.

3. RESULTS AND SIMULATION

3.1. Obtained results

To evaluate the performance of machine learning algorithms, we relied on metrics from the scikit-learn library [44] using mean absolute error (MAE) [45], [46], root mean squared error (RMSE) [47], [48], and R-squared (R2) [49], [50]. They are defined as (1)-(3):

$$MAE(y,\hat{y}) = \frac{1}{n}\sum_{i=0}^{n-1}|y_i - \hat{y}_i| \quad (1)$$

$$R^2(y,\hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{\sum_{i=0}^{n}(y_i-\bar{y})^2} \quad (2)$$

$$RMSE(y,\hat{y}) = \left[\frac{1}{n}\sum_{i=0}^{n-1}(y_i - \hat{y}_i)^2\right]^{\frac{1}{2}} \quad (3)$$

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

MAE and RMSE measure absolute prediction error. Smaller numbers indicate higher prediction performance for two three metrics. The value of R^2 ranges from 0 to 1, with values closer to 1 indicating a better fit of the regression model.

Once the data is prepared for model training, machine learning algorithms usually have many adjustable values or parameters, referred to as hyperparameters, that can influence the model's performance. Therefore, it is necessary to tune these hyperparameters to optimize the model's ability to solve the machine learning problem. To achieve this, we utilized the grid search method, which involves exhaustively searching through a manually specified subset of the hyperparameter space of a learning algorithm. In this approach, a set of values is established for each hyperparameter, and the one that yields the highest accuracy is chosen. Table 1 presents the various hyperparameters used for each model.

Table 2 lists the performance metrics of each machine learning models. The results show that for the exception of linear regression model, random forest regressor gets better results (R-Squared of 0.98) and takes longer time to train, followed by decision tree regressor (score of 0.97) with better results and less training time, then gradient boosting regressor (score of 0.97), but with more training time, and finally K-neighbors regressor, linear regression. Therefore, we can confirm that random forest regressor model is a type of ensemble learning method that combines multiple decision trees to generate predictions. It is known for its ability to

handle large datasets with high-dimensional features, while avoiding overfitting. Therefore, it is a robust and reliable model for traffic flow prediction.

Table 1. Hyperparameters values used in the grid search method for each model

Model	Hyperparameters
Random forest regressor	(*max_depth*=30, *max_features*='auto', *min_samples_leaf*=3, *min_samples_split*=6, *n_estimators*=200)
Decision tree regressor	(*splitter*='best', *max_depth*=45, *min_samples_split*=5, *min_samples_leaf*=2, *max_features*='auto')
K neighbors regressor	(*n_neighbors*=6, algorithm='brute', weights='distance')
Gradient boosting regressor	(*learning_rate*=0.8, *n_estimators*=1200, *max_depth*=10)

Table 2. Comparison of machine learning models performance metrics

Model	RMSE	R2	MAE
Linear regression	511.908	0.5906	318.717
Random forest regressor	111.844	0.980	54.826
Decision tree regressor	134.135	0.971	63.209
Gradient boosting regressor	125.709	0.975	63.203
K neighbors regressor	147.039	0.966	71.713

In order to evaluate the cost-benefit of implementing the machine learning models proposed in this study, we conducted experiments to determine the average training time for each model. These experiments were carried out using the Google Collaboratory platform [51], which allowed us to perform the necessary computations in a cloud-based environment. For each of the five machine learning models tested in this study, we recorded the time it took to train the model on the UK National Road Traffic Estimation Statistics Publication's Road dataset, which was used for both training and testing. The training time was measured in seconds, and the average training time for each model was calculated by taking the mean of the training times across all experiments. Figure 4 presents the results of our experiments in the form of a bar chart, with each bar representing the average training time for a particular machine learning model. As can be seen from the Figure 5, the training time varies widely across the different models, with the decision tree model having the shortest average training time and the neural network model having the longest.

Figure 6 shows a comparison between the actual data and the predicted data using five different machine learning models. Figure 6(a) shows the results of linear regression, Figure 6(b) shows the results of random forest regressor, Figure 6(c) shows the results of decision tree regressor, Figure 6(d) shows the results of gradient boosting regressor, and Figure 6(e) shows the results of K-neighbors regressor.

In each subfigure, the orange line represents the actual traffic flow data, while the blue line represents the predicted traffic flow data. The x-axis represents the time period while the y-axis represents the density values for each segment of the intersection. The results show that the random forest regressor and gradient boosting regressor models outperformed the other models, with the lowest prediction error and the highest accuracy. This comparison was based on different evaluation metrics, such as mean absolute error (MAE), mean squared error (MSE), and coefficient of determination (R-squared). These metrics were computed by comparing the actual data with the predicted data using the test split (20%) of the dataset. The results were then plotted in Figure 6 to provide a visual representation of the performance of each model.

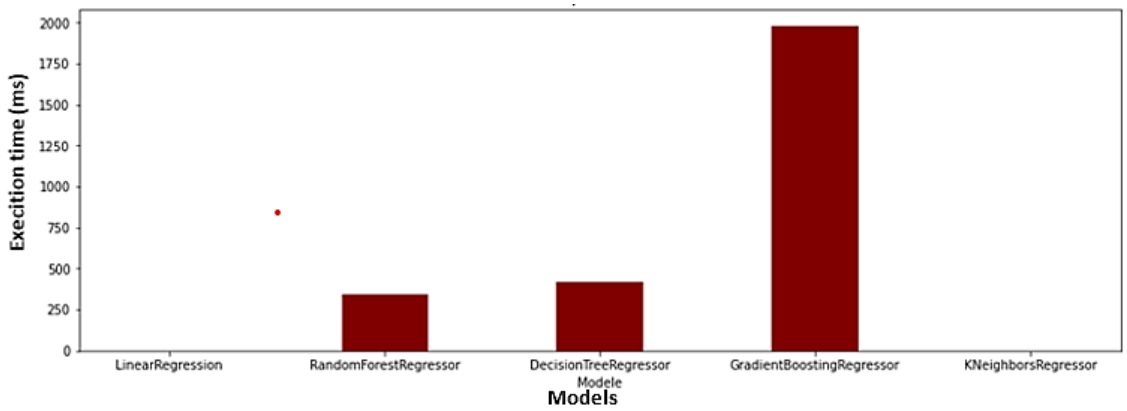

Figure 5. Machine learning model's execution time

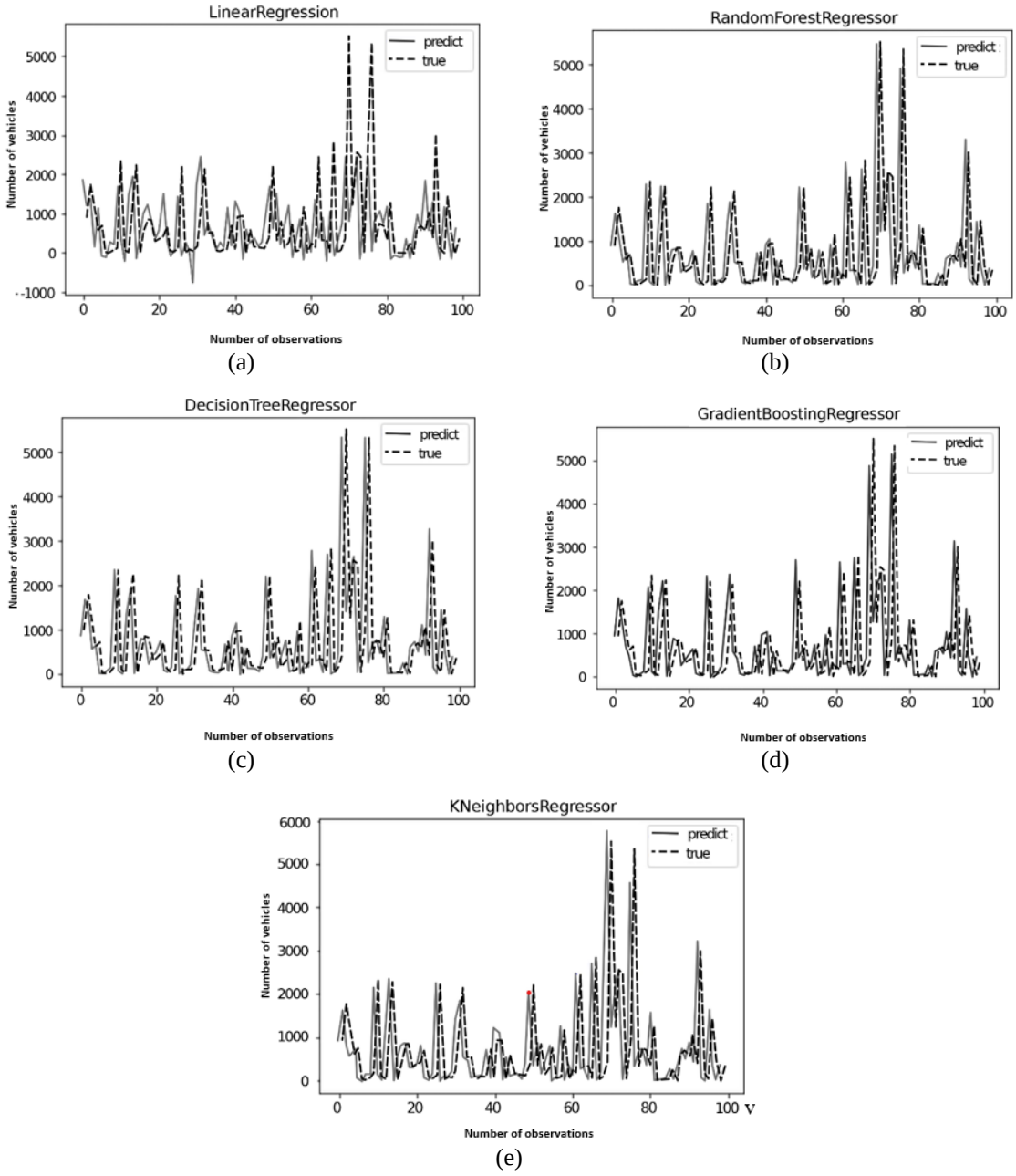

Figure 6. Comparison between the actual data and the predicted data: (a) linear regression, (b) random forest regressor, (c) decision tree regressor, (d) gradient boosting regressor, and (e) k neighbor's regressor

The evaluation of any model is a critical step in determining its effectiveness in solving the targeted problem. In the case of our proposed system aimed at reducing traffic congestion, the evaluation process is crucial to understand its performance and efficiency. To conduct this evaluation, we will analyze the predicted density values for each segment of an intersection before and after the application of the reduction function. By comparing these values, we can determine the extent to which our proposed system has been successful in reducing traffic congestion. Table 3 showcases the results of this analysis, highlighting the effectiveness of our reduction function in minimizing traffic congestion at intersections.

3.2. Traffic simulation

Simulation is a valuable tool for comprehending real-world scenarios, and traffic simulation is widely employed to analyze road traffic congestion. Numerous studies have employed simulation models to simulate traffic congestion, with diverse traffic models accessible, predominantly comprising microscopic, macroscopic, and mesoscopic models. Microscopic models consider the interaction of individual vehicles,

encompassing parameters such as driver behavior, vehicle position, velocity, acceleration, and more. Conversely, macroscopic models analyze the collective behavior of traffic flows, while mesoscopic models integrate features of both macroscopic and microscopic models, describing the impact of nearby vehicles and approximating the cumulative temporal and spatial traffic behavior. In this scholarly paper, we designed and implemented a traffic simulation in Python employing Pygame and a microscopic module (as illustrated in Figure 6 to assess the duration of red and green lights at an intersection. At each time step, the real-time simulation module defines the state of the traffic light phase (red, green) and the duration of each phase.

Table 3. Results obtained from the prediction made for each segment of one intersection with and without the application of the reduction function

Segment	Density without the function	Density with the function
0	378	90
1	666	274
2	1690	806
3	1793	850

Figure 7 illustrate a microscopic model in which, each vehicle in the microscope model is assigned a number i. The i^{th} vehicle follows the (i-1)th vehicle. For the i^{th} vehicle, let x_i be the position on the road, v_i be the velocity, and l_i be the length. And that goes for all vehicles.

$$S_i = x_i - x_{i-1} - l_i \tag{4}$$

$$\Delta v_i = v_i - v_{i-1} \tag{5}$$

The vehicle-to-vehicle (V2V) distance will be represented by S_i and Δv_i the velocity difference between the i^{th} vehicle and the vehicle in front of it (vehicle number i-1). Figure 8 provides a graphical representation of an intersection during the simulation model, illustrating the layout of the road and the positioning of traffic lights. The intersection is divided into several segments, with each segment representing a different part of the road, such as a straight lane or a turning lane. The simulation model takes into account the movement of various types of vehicles, such as cars and trucks, and their interaction with other vehicles and the traffic lights. The model also considers the effect of the reduction function on traffic density and adjusts the traffic light timings accordingly. Overall, the representation of the intersection in Figure 8 provides a clear understanding of the simulation model and how it factors in various elements to optimize traffic flow.

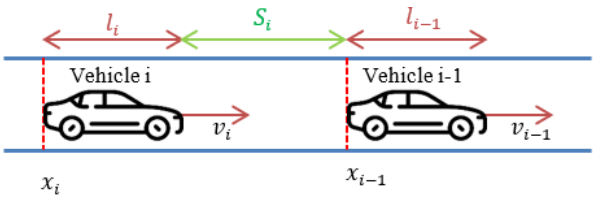

Figure 7. Illustration of a microscopic model

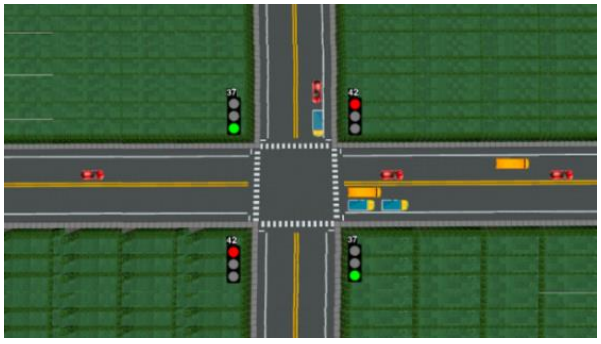

Figure 8. Representation of an intersection during the simulation

This simulation evaluates the optimal timing for the green and red signals at an intersection by using timers on each light to indicate when the light will change. Various types of vehicles are generated and their movements are influenced by the signals and surrounding vehicles. The control system applies prediction and density reduction functions to determine signal plant values, which are then sent to the traffic controller to adjust the traffic lights accordingly. The results of this simulation demonstrate the effectiveness of the proposed system in significantly reducing traffic congestion.

4. CONCLUSION

The focus of this paper was to develop a machine learning-based system that can predict traffic flow at intersections. To achieve this, we implemented several machine learning models and compared their performance. The results showed that the random forest regressor model outperformed the others with an R-squared and EV score of 0.98. The random forest regressor model is a tree-based ensemble algorithm that works by constructing a multitude of decision trees and then outputs the mean prediction of the individual trees. This algorithm is known for its high accuracy, but it also takes more processing time due to its complexity. The decision tree regressor model had a score of 0.97, which is slightly lower than the random forest regressor, but it provided a good balance between measurement of results and training time. The gradient boosting regressor model showed a similar performance to the decision tree regressor but took more processing time. The K neighbors regressor and linear regression models had relatively lower scores, but their performance did not differ significantly from the others. To validate the effectiveness of the proposed system, we simulated the traffic flow at intersections using a microscopic model. The model used a traffic density reduction function to adjust the green and red-light duration, which helped reduce traffic congestion significantly. The efficiency of the suggested system in decreasing traffic congestion provides justification for its deployment at key intersections. However, to enhance the system further, we aim to incorporate additional technologies. One potential area of improvement involves the data used to train the models. Although the dataset used in this study was obtained from various traffic sensors, we intend to explore the use of other data sources, such as traffic camera images, to improve the models' precision. Additionally, we plan to integrate real-time data feeds into the system, enabling it to make real-time decisions, further enhancing its efficacy in managing traffic flow at intersections. In conclusion, the proposed system demonstrates promising results in forecasting traffic flow at intersections and reducing traffic congestion. Although the random forest regressor model proved to be the most precise, the decision tree regressor offered a good balance between accuracy and training time. This system has the potential to be a valuable tool for managing traffic at major intersections, and we anticipate further development and improvement in the future.

REFERENCES

[1] A. Boukerche, N. Aljeri, K. Abrougui, and Y. Wang, "Towards a secure hybrid adaptive gateway discovery mechanism for intelligent transportation systems," *Security and Communication Networks*, vol. 9, no. 17, pp. 4027–4047, Nov. 2016, doi: 10.1002/sec.1586.

[2] K. Abrougui, A. Boukerche, and R. W. N. Pazzi, "Design and evaluation of context-aware and location-based service discovery protocols for vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 717–735, Sep. 2011, doi: 10.1109/TITS.2011.2159377.

[3] M. Bani Younes and A. Boukerche, "A performance evaluation of an efficient traffic congestion detection protocol (ECODE) for intelligent transportation systems," *Ad Hoc Networks*, vol. 24, pp. 317–336, Jan. 2015, doi: 10.1016/j.adhoc.2014.09.005.

[4] C. Rezende, A. Boukerche, H. S. Ramos, and A. A. F. Loureiro, "A reactive and scalable unicast solution for video streaming over VANETs," *IEEE Transactions on Computers*, vol. 64, no. 3, pp. 614–626, Mar. 2015, doi: 10.1109/TC.2014.2308176.

[5] N. Aljeri, K. Abrougui, M. Almulla, and A. Boukerche, "A performance evaluation of load balancing and QoS-aware gateway discovery protocol for VANETs," in *2013 27th International Conference on Advanced Information Networking and Applications Workshops*, Mar. 2013, pp. 90–94, doi: 10.1109/WAINA.2013.232.

[6] C. Chen, B. Liu, S. Wan, P. Qiao, and Q. Pei, "An edge traffic flow detection scheme based on deep learning in an intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1840–1852, Mar. 2021, doi: 10.1109/TITS.2020.3025687.

[7] M. Okawa, H. Kim, and H. Toda, "Online traffic flow prediction using convolved bilinear poisson regression," in *2017 18th IEEE International Conference on Mobile Data Management (MDM)*, May 2017, pp. 134–143, doi: 10.1109/MDM.2017.27.

[8] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transportation Research Part C: Emerging Technologies*, vol. 62, pp. 21–34, Jan. 2016, doi: 10.1016/j.trc.2015.11.002.

[9] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: Deep belief networks with multitask learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2191–2201, Oct. 2014, doi: 10.1109/TITS.2014.2311123.

[10] J. Guo, Z. Wang, and H. Chen, "On-line multi-step prediction of short term traffic flow based on GRU neural network," in *Proceedings of the 2nd International Conference on Intelligent Information Processing*, Jul. 2017, pp. 1–6, doi: 10.1145/3144789.3144804.

[11] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning*. New York, NY: Springer New York, 2009. doi: 10.1007/978-0-387-84858-7.

[12] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*, 1st edition. Cambridge University Press, 2014.

[13] J. Wang and A. Boukerche, "The scalability analysis of machine learning based models in road traffic flow prediction," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, Jun. 2020, pp. 1–6, doi: 10.1109/ICC40277.2020.9148964.

[14] P. Sun, N. Aljeri, and A. Boukerche, "Machine learning-based models for real-time traffic flow prediction in vehicular networks," *IEEE Network*, vol. 34, no. 3, pp. 178–185, May 2020, doi: 10.1109/MNET.011.1900338.

[15] J. Wang, M. R. Pradhan, and N. Gunasekaran, "Machine learning-based human-robot interaction in ITS," *Information Processing and Management*, vol. 59, no. 1, Jan. 2022, doi: 10.1016/j.ipm.2021.102750.

[16] Y. Chen *et al.*, "When traffic flow prediction and wireless big data analytics meet," *IEEE Network*, vol. 33, no. 3, pp. 161–167, May 2019, doi: 10.1109/MNET.2018.1800134.

[17] I. Slimani *et al.*, "Automated machine learning: the new data science challenge," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 4, pp. 4243–4252, Aug. 2022, doi: 10.11591/ijece.v12i4.pp4243-4252.

[18] J. Rice and E. Van Zwet, "A simple and effective method for predicting travel times on freeways," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 3, pp. 200–207, Sep. 2004, doi: 10.1109/TITS.2004.833765.

[19] X. Zhang and J. A. Rice, "Short-term travel time prediction," *Transportation Research Part C: Emerging Technologies*, vol. 11, no. 3–4, pp. 187–210, Jun. 2003, doi: 10.1016/S0968-090X(03)00026-3.

[20] A. G. Hobeika and C. K. Kim, "Traffic-flow-prediction systems based on upstream traffic," in *Proceedings of VNIS'94 - 1994 Vehicle Navigation and Information Systems Conference*, 1994, pp. 345–350, doi: 10.1109/VNIS.1994.396815.

[21] J. Kwon, B. Coifman, and P. Bickel, "Day-to-day travel-time trends and travel-time prediction from loop-detector data," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1717, no. 1, pp. 120–129, Jan. 2000, doi: 10.3141/1717-15.

[22] L. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009, doi: 10.4249/scholarpedia.1883.

[23] B. L. Smith, B. M. Williams, and R. Keith Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 4, pp. 303–321, Aug. 2002, doi: 10.1016/S0968-090X(02)00009-8.

[24] H. Chang, Y. Lee, B. Yoon, and S. Baek, "Dynamic near-term traffic flow prediction: system-oriented approach based on past experiences," *IET Intelligent Transport Systems*, vol. 6, no. 3, 2012, doi: 10.1049/iet-its.2011.0123.

[25] Z. Zheng and D. Su, "Short-term traffic volume forecasting: A k-nearest neighbor approach enhanced by constrained linearly sewing principle component algorithm," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 143–157, Jun. 2014, doi: 10.1016/j.trc.2014.02.009.

[26] D. Xia, H. Li, B. Wang, Y. Li, and Z. Zhang, "A map reduce-based nearest neighbor approach for big-data-driven traffic flow prediction," *IEEE Access*, vol. 4, pp. 2920–2934, 2016, doi: 10.1109/ACCESS.2016.2570021.

[27] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, and L. D. Han, "Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6164–6173, Apr. 2009, doi: 10.1016/j.eswa.2008.07.069.

[28] Z. Mingheng, Z. Yaobao, H. Ganglong, and C. Gang, "Accurate multisteps traffic flow prediction based on SVM," *Mathematical Problems in Engineering*, vol. 2013, pp. 1–8, 2013, doi: 10.1155/2013/418303.

[29] W.-C. Hong, Y. Dong, F. Zheng, and C.-Y. Lai, "Forecasting urban traffic flow by SVR with continuous ACO," *Applied Mathematical Modelling*, vol. 35, no. 3, pp. 1282–1291, Mar. 2011, doi: 10.1016/j.apm.2010.09.005.

[30] M. Duo, Y. Qi, G. Lina, and E. Xu, "A short-term traffic flow prediction model based on EMD and GPSO-SVM," in *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Mar. 2017, pp. 2554–2558. doi: 10.1109/IAEAC.2017.8054485.

[31] S. Oh, Y.-J. Byon, K. Jang, and H. Yeo, "Short-term travel-time prediction on highway: a review of the data-driven approach," *Transport Reviews*, vol. 35, no. 1, pp. 4–32, Jan. 2015, doi: 10.1080/01441647.2014.992496.

[32] F. Jabot, "Why preferring parametric forecasting to nonparametric methods?," *Journal of Theoretical Biology*, vol. 372, pp. 205–210, May 2015, doi: 10.1016/j.jtbi.2014.07.038.

[33] A. Cleeremans, D. Servan-Schreiber, and J. L. McClelland, "Finite state automata and simple recurrent networks," *Neural Computation*, vol. 1, no. 3, pp. 372–381, Sep. 1989, doi: 10.1162/neco.1989.1.3.372.

[34] B. A. Pearlmutter, "Learning state space trajectories in recurrent neural networks," *Neural Computation*, vol. 1, no. 2, pp. 263–269, Jun. 1989, doi: 10.1162/neco.1989.1.2.263.

[35] M. I. Jordan, "Attractor dynamics and parallelism in a connectionist sequential machine," in *IEEE Computer Society Neural Networks Technology Series*, 1990, pp. 112–127.

[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[37] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000, doi: 10.1162/089976600300015015.

[38] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, Nov. 2016, pp. 324–328. doi: 10.1109/YAC.2016.7804912.

[39] R. Jimenez-Moreno, J. E. M. Baquero, and L. A. Rodriguez Umaña, "Ambulance detection for smart traffic light applications with fuzzy controller," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 5, pp. 4876–4882, Oct. 2022, doi: 10.11591/ijece.v12i5.pp4876-4882.

[40] A. F. Abbas, U. U. Sheikh, F. T. AL-Dhief, and M. N. H. Mohd, "A comprehensive review of vehicle detection using computer vision," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, no. 3, pp. 838–850, Jun. 2021, doi: 10.12928/telkomnika.v19i3.12880.

[41] P. D. L. Delica, M. R. U. Landicho, J. A. A. Tabliga, L. R. Virtus, and R. Anacan, "Development of an intelligent traffic control system using NI LabVIEW," in *2017 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, Nov. 2017, pp. 17–22, doi: 10.1109/ICCSCE.2017.8284372.

[42] E. I. Abdul Kareem and H. K. Hoomod, "Integrated tripartite modules for intelligent traffic light system," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 3, pp. 2971–2985, Jun. 2022, doi: 10.11591/ijece.v12i3.pp2971-2985.

[43] S. Lawe and R. Wang, "Optimization of traffic signals using deep learning neural networks," in *Australasian Joint Conference on Artificial Intelligence*, 2016, pp. 403–415, doi: 10.1007/978-3-319-50127-7_35.

[44] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.

[45] Q. Chen, Y. Song, and J. Zhao, "Short-term traffic flow prediction based on improved wavelet neural network," *Neural Computing and Applications*, vol. 33, no. 14, pp. 8181–8190, Jul. 2021, doi: 10.1007/s00521-020-04932-5.
[46] L. Cai, K. Janowicz, G. Mai, B. Yan, and R. Zhu, "Traffic transformer: capturing the continuity and periodicity of time series for traffic forecasting," *Transactions in GIS*, vol. 24, no. 3, pp. 736–755, Jun. 2020, doi: 10.1111/tgis.12644.
[47] C. Luo *et al.*, "Short-term traffic flow prediction based on least square support vector machine with hybrid optimization algorithm," *Neural Processing Letters*, vol. 50, no. 3, pp. 2305–2322, Dec. 2019, doi: 10.1007/s11063-019-09994-8.
[48] Y. Li and C. Shahabi, "A brief overview of machine learning methods for short-term traffic forecasting and future directions," *SIGSPATIAL Special*, vol. 10, no. 1, pp. 3–9, Jun. 2018, doi: 10.1145/3231541.3231544.
[49] A. Boukerche, Y. Tao, and P. Sun, "Artificial intelligence-based vehicular traffic flow prediction methods for supporting intelligent transportation systems," *Computer Networks*, vol. 182, Dec. 2020, doi: 10.1016/j.comnet.2020.107484.
[50] A. Boukerche and J. Wang, "A performance modeling and analysis of a novel vehicular traffic flow prediction system using a hybrid machine learning-based model," *Ad Hoc Networks*, vol. 106, Sep. 2020, doi: 10.1016/j.adhoc.2020.102224.
[51] E. Bisong, *Building machine learning and deep learning models on Google cloud platform*. Berkeley, CA: Apress, 2019, doi: 10.1007/978-1-4842-4470-8.

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, machine and deep learning, 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. He 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. She can be contacted at email: arafalia@yahoo.com.