İSTANBUL UNIVERSITY CERRAHPAŞA

ORIGINAL ARTICLE
Electrica 2022; 22(2): 226-236

Impact Analysis of COVID-19 Pandemic on Istanbul Traffic with Big Data Tools

Uğur Alcan, Fırat Kaçar

Department of Electrical-Electronics Engineering, İstanbul University-Cerrahpaşa, Faculty of Engineering, İstanbul, Turkey

Cite this article as: U. Alcan and F. Kaçar, "Impact analysis of COVID-19 pandemic on istanbul traffic with big data tools", *Electrica*., 22(2), 226-236, 2022.

ABSTRACT

With the internet brought along by technology, people have started to produce data in almost all their jobs. We create a huge data source with many activities we cannot count, such as sending messages on Whatsapp, sharing photos on Instagram, searching in Google, and sending electronic mails (email) and this process is repeated every single day. Such dense and different data also lead to information garbage. Analyzing this dump with traditional technologies has been another problem. Big companies that are interested to analyze this mass information, analyze the behavior of their customers, and determine their strategies according to the results obtained have come up with the concept of big data. Big data are the form of the data we obtain from different sources such as social media shares, sensor data, photo archives, call records obtained from Global System for Mobile Communications (GSM) operators, and search engine statistics, into a meaningful and processable form [1]. In this study, the effect of the coronavirus disease 2019 pandemic, which is an important problem of today, on Istanbul traffic has been examined by using the power of big data technologies. In this context, the hourly traffic index of the 2020 dataset which has openly been published by Istanbul Metropolitan Municipality [2], and the curfew time dataset is discussed. Apache Spark, a new generation data processing tool, has been used in the analysis of these datasets. With Apache Spark, first, general analysis of the Istanbul traffic index data for 2020 has been carried out, and then, the data obtained have been checked whether it is associated with the curfew time dataset and impact analysis has been performed. Elasticsearch has been utilized to keep the processed data, and Kibana has been used for data visualization. At the end of the study, machine learning applications on traffic density have been enhanced using Apache Spark's machine learning library, Application Programming Interface (API) with logistic regression, decision trees, random forest, gradient-boosted tree-based OneVsRest, and linear support vector machine-based OneVsRest methods.

Index Terms—Apache spark, big data, elasticsearch, kibana, traffic analysis

I. INTRODUCTION

With the coronavirus disease 2019 (COVID-19) spreading globally since its first appearance in China in December 2019, it has become a pandemic with an unprecedented impact on all countries. Since the transmission mechanism of the virus was largely unknown in the early stages, governments tried to combat the pandemic by taking different measures. The most important of these measures was to impose restrictions on the public on various issues. The majority of the world's population was affected by these restrictions and had to change, directly or indirectly, in most of their daily activities. During the peak of the COVID-19 pandemic, many countries imposed curfews at certain intervals. Although these restrictions have changed over time, travel within the country has continued in various ways [3].

In this study, the effect of the COVID-19 virus and curfews on Istanbul traffic was examined. In the study, the 2020 Istanbul hourly traffic index [2] is obtained from Istanbul Metropolitan Municipality (IMM) and the 2020 curfew time datasets were used.

First of all, using Apache Spark, one of the big data processing tools in these data sets, hourly traffic index data set for 2020 was used to analyze the traffic situation of 1 year in criteria such as day, month, hour. Later, the criterion of whether there is a curfew was added to these criteria and the study was deepened.

The results obtained were transferred to Elasticsearch, which is a not only structured query language (NoSQL) database. Kibana was used to visualize the processed data here.

The article was made from the thesis work. The thesis work was defended on 11.01.2022 at Istanbul University-Cerrahpaşa and was approved by the juries.

Corresponding author:
Uğur Alcan

E-mail: uguralcan96@gmail.com

Received: January 14, 2022

Accepted: February 22, 2022

DOI: 10.54614/electrica.2022.210005

In the last part of the study, the following machine learning applications were made on the traffic index by using the machine learning library (Mllib), which is one of the APIs of Apache Spark; logistic regression, decision trees, random forest, gradient boost tree-based OneVsRest, and linear support vector machine-based OneVsRest.

II. MATERIALS AND METHODS

In this study, the effects of the COVID-19 pandemic and the COVID-19-induced curfews on Istanbul traffic were determined. At the end of the study, traffic situation prediction was made with five different methods. The data were processed with Apache Spark and visualized with Kibana. Elasticsearch was used to keep the processed data. In the traffic predicting part, Apache Spark's MLlib is used.

A. Data Used In The Study

The 2020 Istanbul Traffic index dataset used in this study was taken from the open data portal of IMM [2]. The data include the average Istanbul Traffic index values with 1 hour intervals. The traffic index takes values ranging from 1 to 99. The larger the index value, the more concentrated the traffic. In this data set, there are basically two columns, the date and the traffic index value. Before the analysis, two more columns were added to this data set regarding whether there were a curfew and a public holiday.

B. Data Analysis and Visualization

Apache Spark was used in the analysis of the datasets used in the study. Elasticsearch was used to keep the processed data, and Kibana was preferred for the visualization of this data.

1) Apache Spark

Apache Spark is a next-generation data-processing engine capable of in-memory processing of big data. Although Apache Spark is developed with scala, it also allows development with languages such as Java and Python. The most important feature that makes Spark stand out is its speed in data processing due to its in-memory operation.

Apache Spark was used in all analyses and machine-learning applications made within the scope of this study. Java is preferred as the programming language in the developments made with Apache Spark.

Apache Spark contains five different components, namely, Spark SQL, Spark Core, Spark Streaming, Mllib, and Graph X [4,5].

a) Spark Core

Spark Core is the part that forms the basis of all components of spark. It is responsible for activities such as memory management, error management, and task allocations in Spark jobs. The most basic dataset structure of Spark, Resilient Distributed Dataset (RDD), is found in Spark Core [6].

b) Spark SQL

Spark SQL is Spark's library that focuses on structured data. Spark SQL allows you to structure data in semi-structured formats such as json, csv, as DataFrame or Dataset and to throw unique SQL queries on them. Dataset and DataFrame are also distributed datasets like RDD. The most important difference from RDD is that Dataset and DataFrame structures are arranged in rows and columns, similar to the classical relational database [5,7].

c) Spark MLlib

Modeling to make predictions based on data is called machine learning. The component of Spark that contains the libraries required for machine learning is called Spark MLlib. Along with the Apache Spark MLlib library, it has also made it possible to adapt machine learning to big data architecture and make it scalable. Apache Spark MLlib has many methods for machine-learning algorithms such as classification, regression, clustering, and dimensionality reduction [8].

d) Spark Streaming

It is the component of Apache Spark that enables analysis of streaming data. Instant data analysis can be made with spark streaming from sources such as Flume, Nifi, Kafka, and Hdfs. Spark streaming also allows the implementation of machine-learning and GraphX algorithms on streaming data.

e) GraphX

GraphX is Apache Spark's component for graph analysis. It converts RDDs to directional graphs and provides processing.

2) Elasticsearch

Elasticsearch is an open-source search and analysis platform based on the indexing mechanism, developed with java on Apache Lucene. Elasticsearch can work in a distributed architecture. It keeps the data in json format. Elasticsearch is highly preferred, especially in tasks such as querying and text searching [9].

3) Kibana

It is the component that allows end-users to view projects made with ELK software called Kibana Elastic Stack. The name ELK comes from the initials of Elasticsearch, Logstash, and Kibana applications. In projects made with ELK, Elasticsearch is the component that keeps the data and enables search operations, and Logstash is the component that collects, processes, and transports the data. Kibana is used during the visualization of the processed data [10].

C. Machine-Learning Applications

Five methods were used in this study. Logistic regression was used for traffic density estimation due to its easy applicability and widespread use. Decision tree was used because it is one of the successful general-purpose algorithms and random forest method was used to minimize the risk of overfitting problem by creating a model with more than one decision tree and to increase the probability of having a high success rate. In order to see how the binary classification algorithms will show in practice, the gradient boost tree and linear support vector machine-based OneVsRest method are used in addition to these three algorithms. Since classification algorithms will be used, the traffic density is divided into three categories as follows.

$0 < \text{traffic index} \leq 20 \rightarrow 1$: normal level traffic

$20 < \text{traffic index} \leq 50 \rightarrow 2$: moderate level traffic

$50 < \text{traffic index} \leq 100 \rightarrow 3$: heavy traffic

1) Used Machine-Learning Methods

In the study, logistic regression, decision trees, random forest, gradient-boosted tree-based OneVsRest, and linear support vector machine-based OneVsRest methods, which are the most widely used classification methods in machine learning applications, were

used. Month, day of the month, day, hour, curfew, and public holiday are taken as feature columns.

a) Logistic Regression

Logistic regression is a machine-learning algorithm that is mostly used in two-class classification algorithms but can also be generalized to predict problems with more than two classes. In this algorithm, a K-1 logistic model is created in a K class system [11].

b) Decision Tree

Decision trees are machine-learning methods used for classification and regression purposes. With this algorithm, various questions are asked to the data given for training and a model is created according to the answers received. It is frequently preferred in machine-learning applications because it is easier to understand and interpret than other methods [12,13].

c) Random Forest

Random forest algorithm is one of the machine-learning algorithms based on the decision tree method, which can be used for both regression and classification applications. The random forest method divides the data sets into random subsets and trains them in order to overcome the problem of memorizing the data in the decision trees algorithm. In this way, it brings together multiple decision trees and makes predictions for each of them. Random forest algorithm performs the estimation process by taking the average of the real number returned by each tree in the regression, according to the estimation with the most votes in the classification [14].

d) OneVsRest

Some popular classification methods used in machine learning are only used in binary classification situations. Therefore, these methods cannot be applied directly to multi-class problems. To overcome this situation, some methods have been developed to separate multiple classes into many binary classes. One of these methods is the OneVsRest classification method. This method, also known as one against all, is a supervised classification method that provides multi-class classification by giving a method that can effectively perform binary classification as the base classifier [15].

e) Gradient-Boosted Tree

The gradient boost tree is one of the ensemble algorithms used in classification and regression problems. It aims to minimize errors by training decision trees iteratively. In this method, new models are created in each iteration to correct the mistakes made in the previous iterations. In other words, a sequential process is performed in which the later models are dependent on the previous ones [16].

f) Linear Support Vector Machine

Support Vector machines are supervised machine-learning algorithm used in regression and classification problems. It is generally preferred in classification problems. The main purpose of the support vector machine algorithm is to determine the boundary that best separates the n-dimensional space from the training data in order to categorize the data to be estimated correctly [17].

2) Model Evaluation Metrics

There are some metrics to evaluate the success of machine-learning models created with classification algorithms. These metrics are generally obtained with the confusion matrix. The confusion matrix is a structure that shows the number of correctly and incorrectly predicted classes in the model.

TABLE I. CONFUSION MATRIX

		Predicted		
	Classes	$c_{0}\dots c_{k-1}$	c_k	$c_{k+1}\dots c_n$
Actual	c_0 … c_{k-1}	TN	FP	TN
	c_k	FN	TP	FN
	c_{k+1} … c_n	TN	FP	TN

The confusion matrix is given in Table I c_{ij} represents the frequency of estimation of class i as class j. A confusion matrix contains four kinds of classification results according to the target class k. These results are expressed as TP, TN, FP, and FN [18].

TP (True Positive): true prediction of the positive class

$$TP = c_{k,k} \tag{1}$$

TN (True Negative) : true prediction of the negative class

$$TN = \sum_{i,j \in N \setminus \{k\}} c_{ij} \tag{2}$$

FP (False Positive) : wrong prediction of the positive class

$$FP = \sum_{i \in N \setminus \{k\}} c_{ik} \tag{3}$$

FN (False Negative) : wrong prediction of the negative class

$$FN = \sum_{i \in N \setminus \{k\}} c_{ki} \tag{4}$$

The following metrics are generally used when evaluating the success of classification algorithms with the help of the confusion matrix.

Accuracy: it refers to the ratio of correct predictions to all predictions.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

Precision: a measure of the classifier's success in predicting a particular class correctly.

$$Precision_{class} = \frac{TP_{class}}{TP_{class} + FP_{class}} \tag{6}$$

Recall: indicates the classifier's ability to accurately predict a class.

$$Recall_{class} = \frac{TP_{class}}{TP_{class} + FN_{class}} \tag{7}$$

F1 score: it is a measure obtained from the harmonic mean of precision and precision. It shows how precise and powerful the classifier is.

$$F1score_{class} = \frac{2.TP_{class}}{2.TP_{class} + FN_{class} + FP_{class}} \quad (8)$$

III. IMPLEMENTATION AND RESULTS

In this section, first of all, the results of the traffic analysis for 2020 and the effects of the curfews on the traffic situation are shared. In the last part of the section, the results of machine-learning applications designed with logistic regression, random forest, decision tree, gradient boosted tree-based OneVsRest, and linear support vector machine-based OneVsRest are given for traffic density estimation. Spark SQL was used for data analysis and Spark MLlib was used for traffic density estimation.

A. Istanbul Traffic Situation Analysis Results in 2020

When the Istanbul average traffic index analysis for 2020 is done on an hourly basis, the result is as in Fig. 1.

Looking at Fig. 1, it is striking that the traffic is heavy, especially during the return home hours. The heaviest traffic hour for 2020 seems to be 18:00.

When we detail the traffic analysis on an hourly basis and separate it according to whether there is a curfew or not, the result obtained is as seen in Fig. 2.

Looking at Fig. 2, the situation during the hours when there is no curfew is quite similar to the situation in Fig. 1. Again, evening home commute hours were the busiest hours of the traffic index. As expected, during the curfew hours, all hours have very low traffic index values. Since the traffic index values are low between 0 and 6 o'clock in the absence of curfew, the average traffic index values were quite close during this time period, with and without curfew.

When the analysis is made in terms of the days of the week, the result in Fig. 3 is obtained.

Looking at Fig. 3, it is seen that although the average traffic index values of the weekdays are close to each other, Friday, which is the end of the working day, is at the top and the weekend has low values.

When the average traffic index analysis on a day-by-day basis is performed on a day-hour basis, the heat map in Fig. 4 is obtained.

Looking at Fig. 4, it is seen that the hour with the highest average traffic index on all weekdays is 18:00. In addition, it can be concluded that the traffic on weekdays is mostly during the return home hours. On the weekend, it is seen that the traffic is the busiest at 15:00.

When the results obtained in Fig. 4 are separated according to the situation whether there is a curfew or not, the situation in Fig. 5 is obtained. In the absence of a curfew, it is seen that the day-hour traffic conditions during the week are quite similar. On weekdays when there is a curfew, it is seen that the highest traffic is at 21:00. Another striking detail here is that the only day in 2020 that does not have a curfew in all time zones is Wednesday.

When the traffic analysis of Istanbul 2020 is made on a monthly basis, the results in Fig. 6 are obtained.

Looking at Fig. 6, it is seen that the average traffic index values of January and February are close to each other and around 30. In March, this value was measured as 21. If we separate the month of March as before and after the coronavirus (March 11 and before the first case was announced was named as pre-coronavirus, and the situation from March 11 to the end of the month was named as post-coronavirus.), the traffic index before the coronavirus was 31 and after the coronavirus was 16. In April, it seems that the average traffic index value decreased to 10 with the closure of schools, the transition of most companies to work from home and the curfews. The lowest average traffic index in 2020 belongs to April.

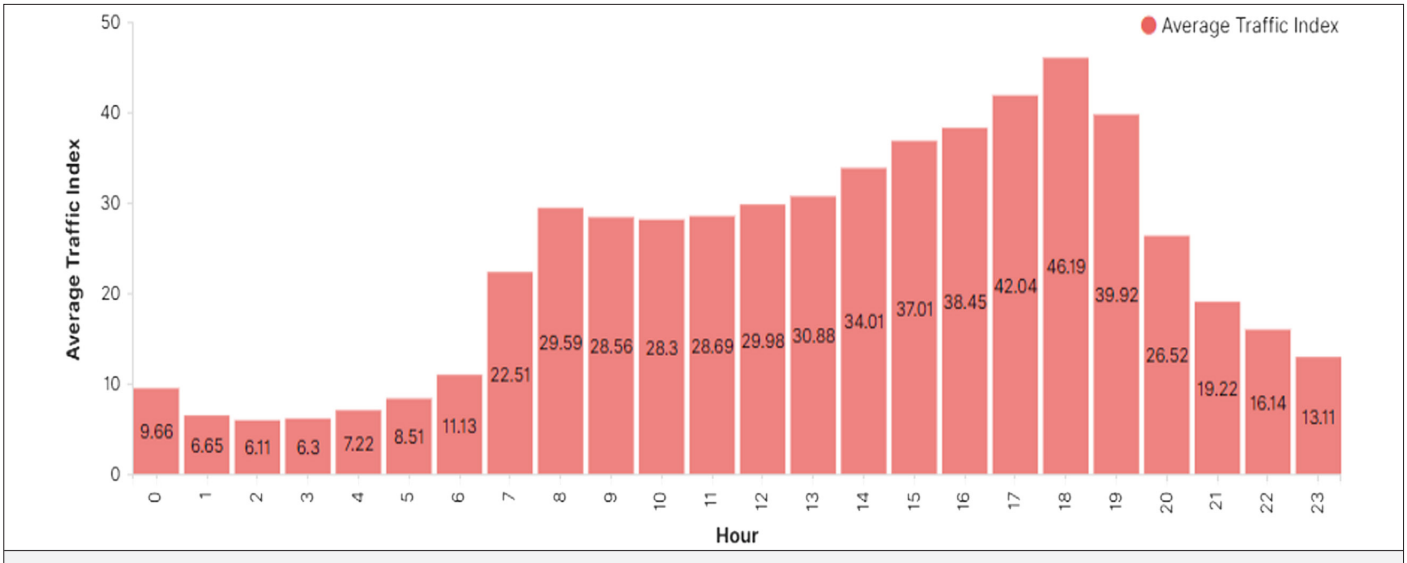

Fig. 1. Istanbul 2020 hourly average traffic index column chart.

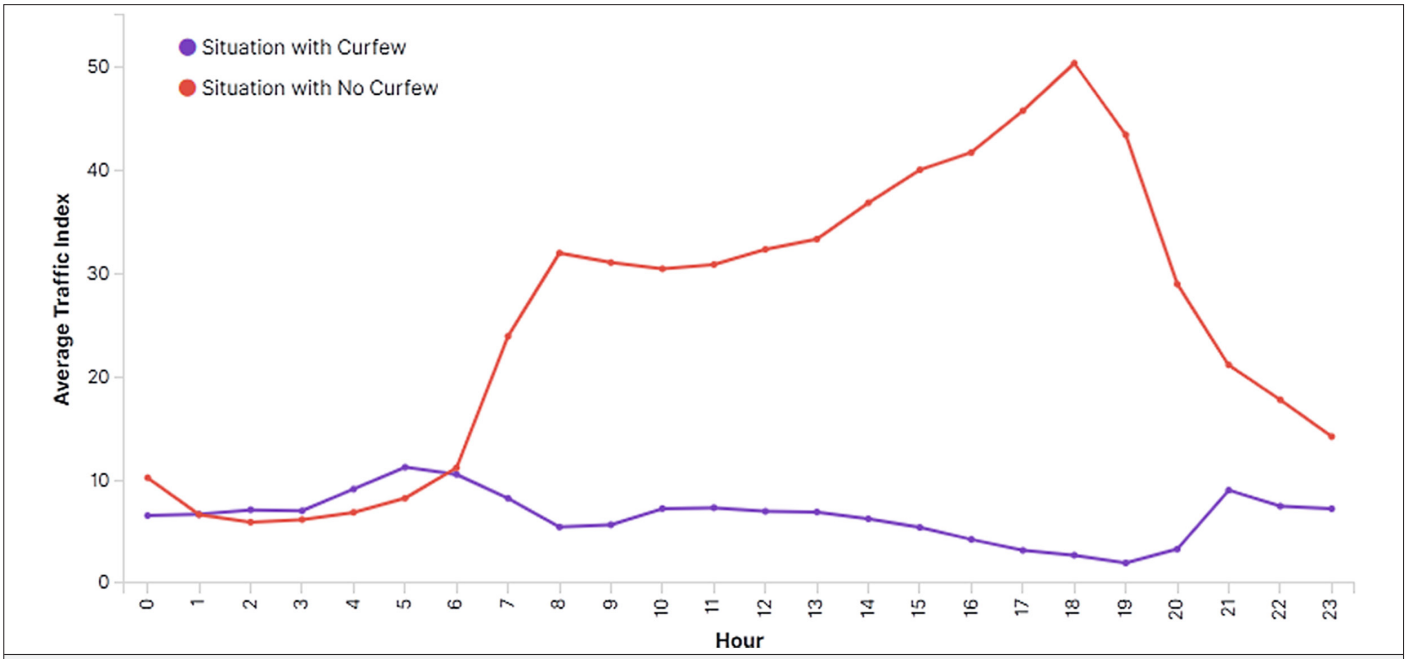

Fig. 2. Graph of the effect of the curfews in Istanbul 2020 on the average traffic index on an hourly basis.

When the monthly based analysis is divided into hourly, the result in Fig. 7 is obtained. When the graph is examined, it is seen that the hour with the busiest traffic in all months is 18. Especially before March C-19; In the transition to post-March C-19, traffic has dropped drastically in almost all time zones. In Fig. 6, it was stated that April had the lowest average traffic index for 2020, looking at Fig. 7, April also had low average traffic index values in all time zones. Traffic index values have increased since June, with 28 traffic index values in September and October, it has reached the state where the traffic has the highest average traffic index value in the months when COVID-19 showed its effect.

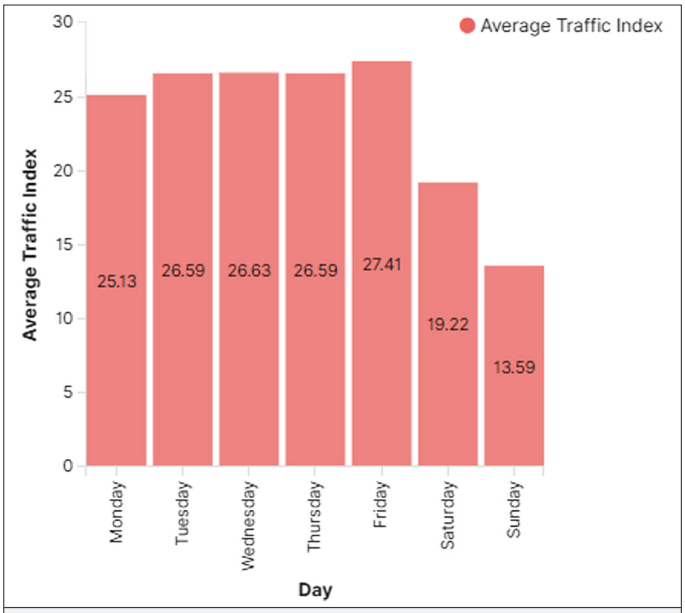

Fig. 3. Average traffic index graph by day of the week for 2020.

Finally, when looking at the average traffic index status of all days for 2020, the heat map in Fig. 8 is obtained.

B. Traffic Prediction Machine-Learning Applications

This part of the study is about the results of machine-learning applications designed for traffic situation prediction. Logistic regression, random forest, decision tree, gradient boosted tree-based OneVsRest, and linear support vector machine-based OneVsRest methods, which are the most widely used classification methods in machine learning applications, were used. Month, day of the month, day, hour, curfew, and whether it is a public holiday are taken as feature columns. 70% of the data was given as train data and 30% as test data.

1) Traffic Situation Prediction with Logistic Regression Method

The confusion matrix in Table II was obtained as a result of estimating the relevant traffic situation with logistic regression.

The model created by logistic regression, in terms of estimating the traffic situation, which we have divided into three categories, it is striking that it fails to predict the heavy traffic. This method yielded no accurate results in estimating heavy traffic conditions. Our model did not achieve successful results in traffic predicting at this level, with 56% precision and 65% recall values for moderate level traffic predicting. The success metrics for traffic estimation of the model created by the logistic regression method are given in Table III.

2) Traffic Situation Prediction with Decision Tree Method

The confusion matrix in Table IV is obtained as a result of the Traffic Condition estimation using the decision tree method.

The model created by the decision tree method seems that achieves a good result with 50% precision and 65% recall in the prediction of heavy traffic situation. This method has achieved a good result with 74% precision and 80% recall in moderate-level traffic predicting. The

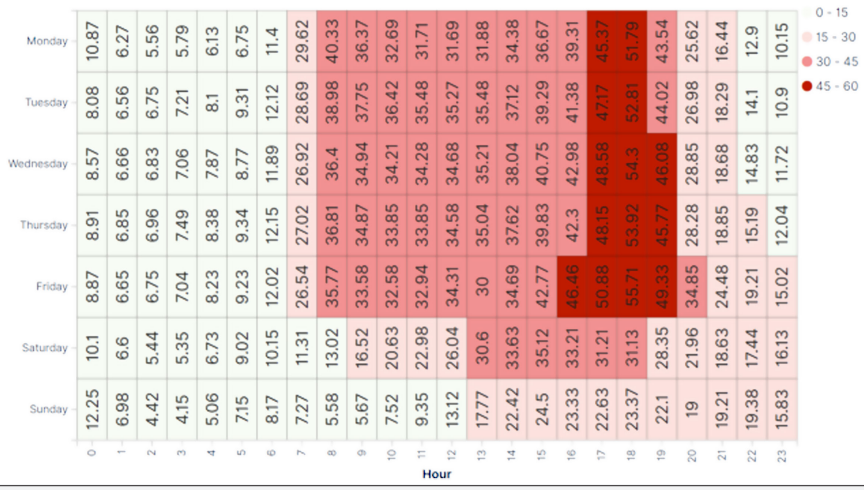

Fig. 4. Average traffic index heat map based on day-hour for 2020.

success metrics of the model created by the decision tree method are given in Table V.

3) Traffic Situation Prediction with Random Forest Method

In order to perform the related machine-learning application with the random forest method, the number of trees was checked up to ten and it was determined that the model had the highest accuracy rate when the number of trees was two. The confusion matrix of the model created with the random forest method is as Table VI.

As can be seen from Table VI, it seems that the model created by the random forest method generally achieved good results. The model created with this method achieved 61% precision and 62% recall in

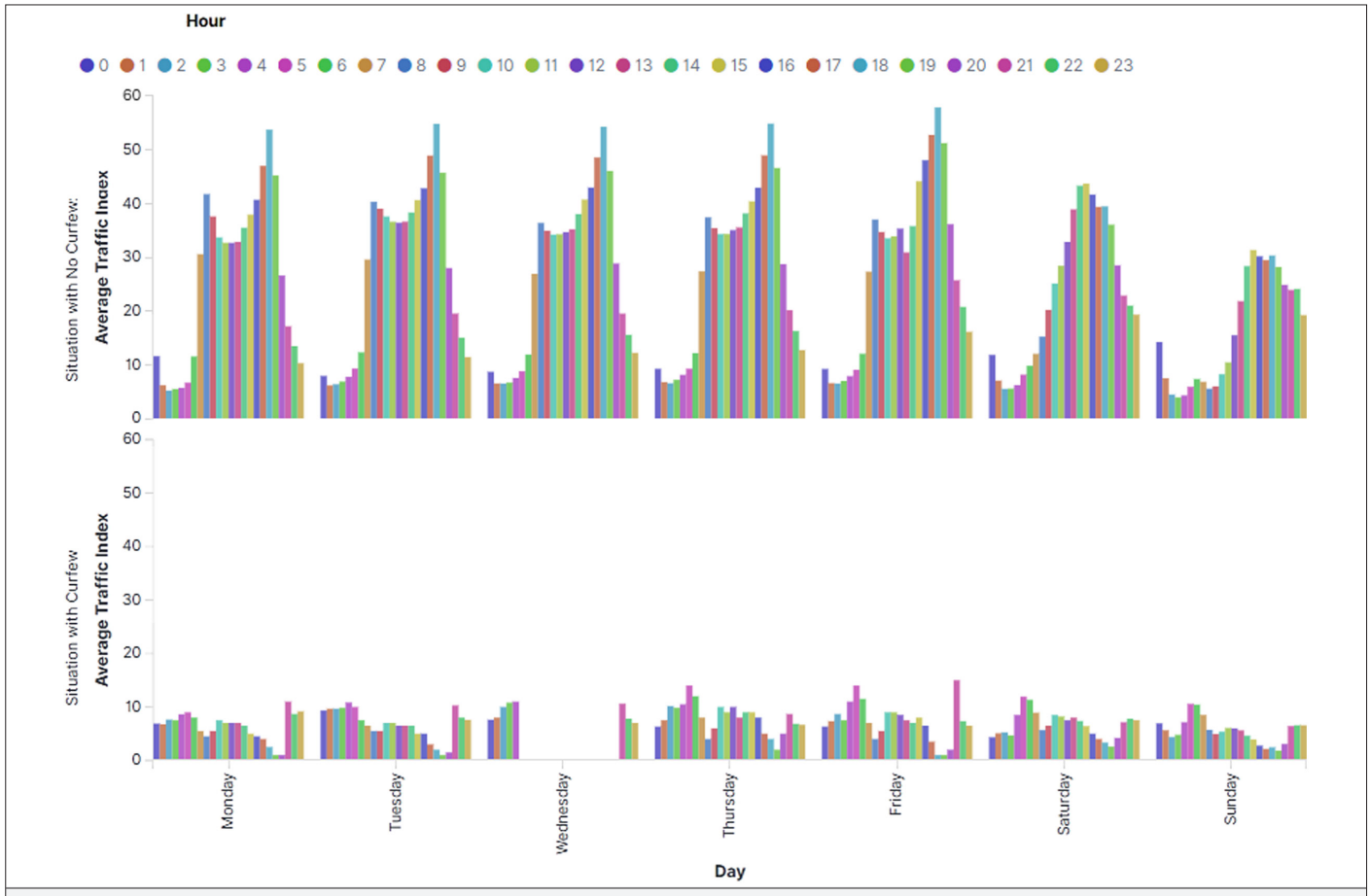

Fig. 5. Graph of the effect of curfews in 2020 on the average traffic index based on day-hour.

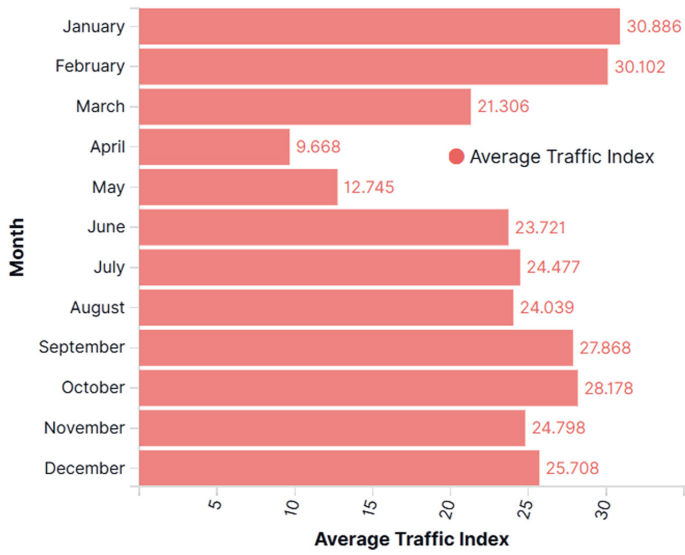

Fig. 6. Average traffic index graph by month for 2020.

heavy traffic prediction. The success metrics of the model created with the random forest method are given in Table VII.

4) Traffic Situation Prediction with Gradient Boosted Tree-based OneVsRest Method

In the previous methods, the traffic situation prediction was carried out directly by the methods that provide multi-class classification. This method, on the other hand, is made by classifying multiple classes with the OneVsRest method of the gradient boost tree method used in binary classification. Table VIII shows the results of the model created with the GBT-based OneVsRest method, with the confusion matrix.

The model created with the GBT-based OneVsRest method seems to achieve quite good results in estimating all traffic situation classes. Heavy traffic predict; Considering that it is more difficult than the estimation of other situations, this method stands out with 82% precision and 70% recall in heavy traffic prediction. The success metrics of the model created with the GBT-based OneVsRest method are given in Table IX.

5) Traffic Situation Prediction with SVM-based OneVsRest Method

The confusion matrix of the model created with the linear support vector machine-based OneVsRest method is given in Table X.

Looking at Table X, it is seen that the results of the model created by the SVM-based OneVsRest method are quite similar to the results of the model created by the logistic regression method. The SVM-based OneVsRest method, like the logistic regression method, did not obtain any accurate results in heavy traffic predicting. The success metrics of the model created with the SVM-based OneVsRest method are given in Table XI.

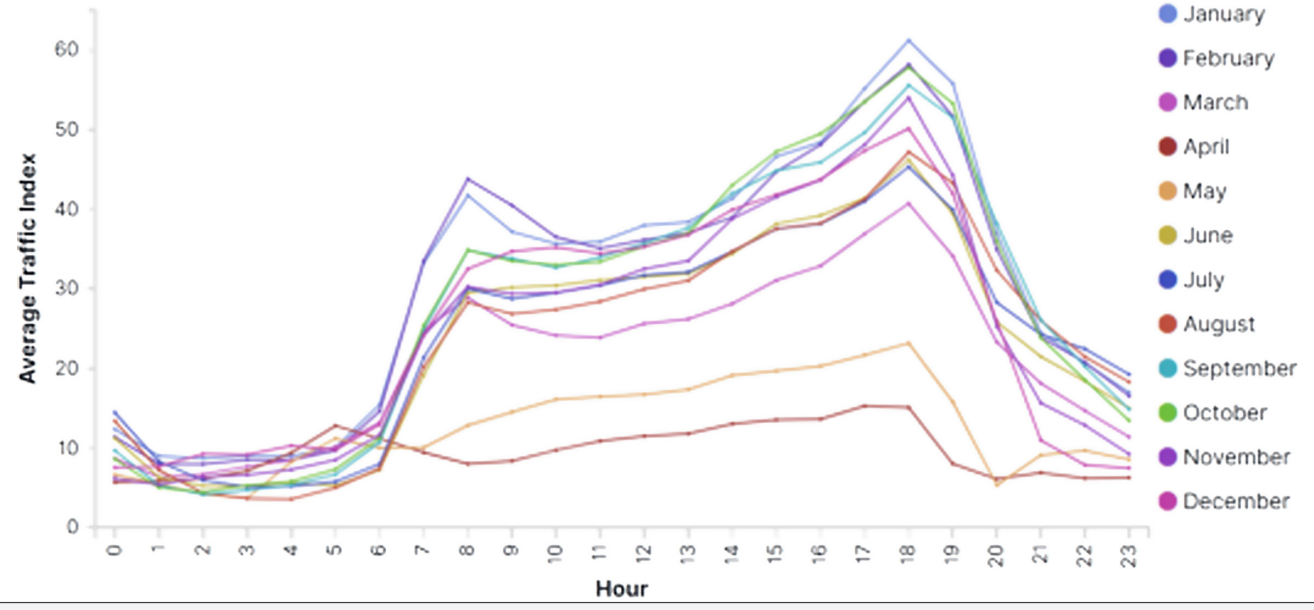

Fig. 7. Istanbul 2020 month-hour based average traffic index graph.

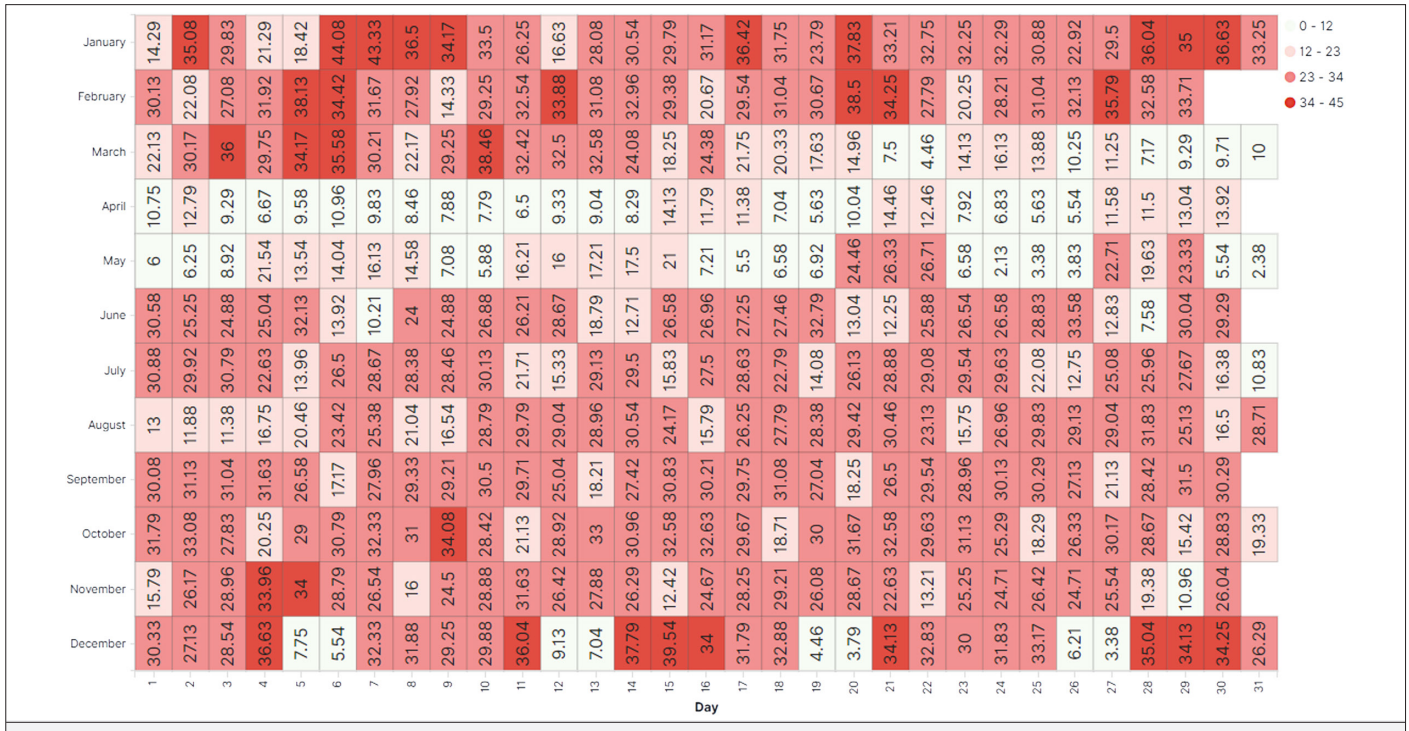

Fig. 8. Istanbul 2020 average traffic index heat map.

TABLE II. CONFUSION MATRIX WITH LOGISTIC REGRESSION METHOD

Traffic Status	Normal Level Traffic	Moderate Level Traffic	Heavy Traffic
Normal Level Traffic	1136	271	0
Moderate Level Traffic	331	619	0
Heavy Traffic	20	214	0

TABLE III. LOGISTIC REGRESSION METHOD SUCCESS METRICS

	True Count	False Count	Precision	Recall	F1 Score
Normal Level Traffic	1136	271	0.76	0.80	0.78
Moderate Level Traffic	619	331	0.56	0.65	0.60
Heavy Traffic	0	234	0	0	0

TABLE IV. CONFUSION MATRIX WITH DECISION TREE METHOD

Traffic Status	Normal Level Traffic	Moderate Level Traffic	Heavy Traffic
Normal Level Traffic	1179	188	40
Moderate Level Traffic	72	767	111
Heavy Traffic	1	80	153

TABLE V. DECISION TREE METHOD SUCCESS METRICS

	True Count	False Count	Precision	Recall	F1 Score
Normal Level Traffic	1179	228	0.94	0.83	0.88
Moderate Level Traffic	767	183	0.74	0.80	0.77
Heavy Traffic	153	81	0.50	0.65	0.56

TABLE VI. CONFUSION MATRIX WITH RANDOM FOREST METHOD

Traffic Status	Normal Level Traffic	Moderate Level Traffic	Heavy Traffic
Normal Level Traffic	1190	182	35
Moderate Level Traffic	77	817	56
Heavy Traffic	1	87	146

TABLE VII. RANDOM FOREST METHOD SUCCESS METRICS

	True Count	False Count	Precision	Recall	F1 Score
Normal Level Traffic	1190	217	0.93	0.84	0.88
Moderate Level Traffic	817	133	0.75	0.86	0.80
Heavy Traffic	146	88	0.61	0.62	0.61

TABLE VIII. CONFUSION MATRIX WITH GBT BASED ONEVSREST METHOD

Traffic Status	Normal Level Traffic	Moderate Level Traffic	Heavy Traffic
Normal Level Traffic	1329	74	4
Moderate Level Traffic	56	863	31
Heavy Traffic	0	70	164

TABLE IX. GBT-BASED ONEVSREST METHOD SUCCESS METRICS

	True Count	False Count	Precision	Recall	F1 Score
Normal Level Traffic	1329	78	0.95	0.94	0.95
Moderate Level Traffic	863	87	0.85	0.90	0.88
Heavy Traffic	164	70	0.82	0.70	0.75

TABLE X. CONFUSION MATRIX WITH SVM-BASED ONEVSREST METHOD

Traffic Status	Normal Level Traffic	Moderate Level Traffic	Heavy Traffic
Normal Level Traffic	1119	288	0
Moderate Level Traffic	281	669	0
Heavy Traffic	17	217	0

TABLE XI. SVM-BASED ONEVSREST METHOD SUCCESS METRICS

	True Count	False Count	Precision	Recall	F1 Score
Normal Level Traffic	1119	288	0.78	0.79	0.79
Moderate Level Traffic	669	281	0.56	0.70	0.62
Heavy Traffic	0	234	0	0	0

VI. CONCLUSIONS

Within the scope of this study, using the 2020 Istanbul traffic index and curfew datasets, 2020 Istanbul traffic analysis was made with Apache Spark. Analyses were made according to day, month, hour, and whether there was a curfew or not. It has been determined that the month with the busiest traffic in 2020 is January, the day is Friday, and the time is 18:00. As expected, low-traffic densities were obtained in the time periods of the curfew. In monthly based analyzes, it has been determined how COVID-19 has caused a change in Istanbul traffic. It has been determined that the traffic density in January and February is close to each other, and in March, the traffic density has decreased since the first coronavirus case in our country. With schools being closed in April, most companies switching to working from home and curfews, traffic density dropped to the bottom and April was determined as the month with the lowest traffic density in 2020. The traffic density, which has increased since June, reached the highest values in September and October, among the months when the effect of COVID-19 was observed. The analysis results obtained were transferred to the NoSQL database and visualization of these data was made with kibana. After the traffic situation analysis of 2020 was done and the results were obtained, machine-learning applications were made for traffic situation prediction with logistic regression, decision trees, random forest, GBT-based OneVsRest, and SVM-based OneVsRest methods using the traffic index dataset of 2020. In the applications, month, day, day of the month, hour, curfew, and whether there is a public holiday or not are taken as feature columns. When the accuracy rates of the five designed models were compared, GBT-based OneVsRest got the best results. The GBT-based OneVsRest method has outstripped other methods, especially in heavy traffic estimation. Random forest and decision tree methods, on the other hand, gave satisfactory results, although they were not as successful as the GBT-based OneVsRest method in terms of accuracy and heavy traffic estimation. As for the logistic regression and SVM-based OneVsRest methods, these methods gave very similar results to each other. Both methods failed to predict the heavy traffic sitution. Accuracy rates of the methods used in the study given in Fig. 9.

Peer-review: Externally peer-reviewed.

Author Contributions: Concept – U.A., F.K.; Design – U.A.; Supervision – F.K.; Materials – U.A.; Data Collection and/or Processing – U.A.; Analysis and/or Interpretation – U.A., F.K.; Literature Review – U.A.; Writing – U.A.; Critical Review – F.K.

Declaration of Interests: The authors declare that they have no competing interest.

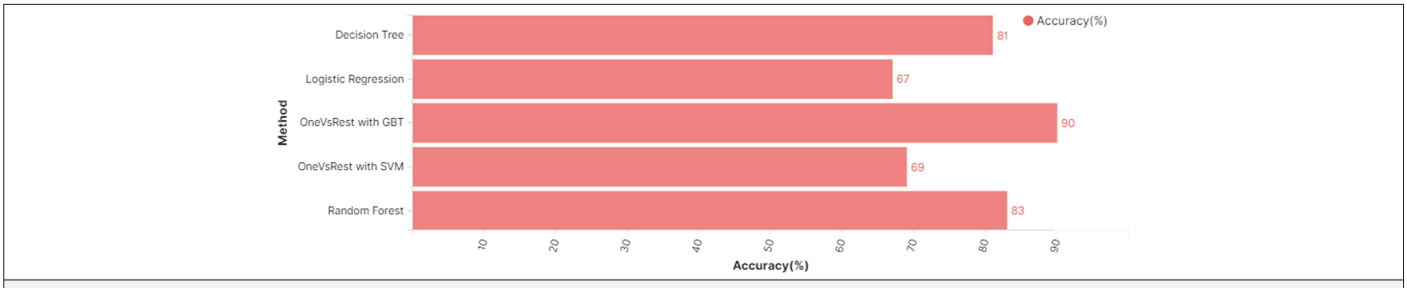

Fig. 9. Comparison of the accuracy rates of the methods used in traffic situation prediction.

Funding: This study received no funding.

REFERENCES

1. E. Aktan, "Büyük veri: Uygulama alanları, Analitiği ve Güvenlik Boyutu," *Bilgi Yönetimi*, vol. 1, no. 1, pp. 1–22, 2018. [CrossRef]
2. Available: https://data.ibb.gov.tr/dataset/trafik-indeks-degeri-web-servisi.
3. S. Parr, B. Wolshon, J. Renne, P. Tuite, and K. Kim, "Traffic impacts of the COVID-19 pandemic: Statewide analysis of social separation and activity restriction," *Nat. Hazards Rev.*, vol. 21, no. 3, 2020. [CrossRef]
4. Available: https://databricks.com/spark/about.
5. F. Bilgin, "Apache spark'a giriş," 2020. Available: https://www.veribilimiokulu.com/apache-sparka-giris/.
6. Available: https://spark.apache.org/docs/2.2.0/rdd-programming-guide.html. [Accessed July 22, 2021].
7. Available: https://spark.apache.org/docs/2.2.0/sql-programming-guide.html#overview. [Accessed July 23, 2021].
8. Available: https://spark.apache.org/docs/2.2.0/ml-classification-regression.html. [Accessed July 23, 2021].
9. K. Doğan, "Elasticsearch Nedir," 2019. Available: https://medium.com/@kdrcandogan/elasticsearch-nedir-45d237c29b26.
10. Available: https://www.elastic.co/what-is/kibana. [Accessed August 15, 2021].
11. Available: https://spark.apache.org/docs/2.2.0/ml-classification-regression.html#logistic-regression. [Accessed August 07, 2021].
12. Available: https://spark.apache.org/docs/2.2.0/mllib-decision-tree.html. [Accessed August 07, 2021].
13. M. F. Akca, "Karar Ağaçları," 2020. Available: https://medium.com/deep-learning-turkiye/karar-a%C4%9Fa%C3%A7lar%C4%B1-makine-%C3%B6%C4%9Frenmesi-serisi-3-a03f3ff00ba5.
14. Available: https://spark.apache.org/docs/2.2.0/mllib-ensembles.html#random-forests. [Accessed August 14, 2021].
15. "One-vs-rest strategy for multi-class classification," 2020. Available: https://www.geeksforgeeks.org/one-vs-rest-strategy-for-multi-class-classification/.
16. V. Kurama, "Gradient boosting in classification: Not a black box anymore!," 2019. Available: https://blog.paperspace.com/gradient-boosting-for-classification/.
17. M. F. Akca, "Nedir bu destek Vektör Makineleri?," 2020. Available: https://medium.com/deep-learning-turkiye/nedir-bu-destek-vekt%C3%B6r-makineleri-makine-%C3%B6%C4%9Frenmesi-serisi-2-94e576e4223e.
18. S. Abraham, C. Huynh, and H. Vu, "Classification of soils into hydrologic groups using machine learning," *Data*, vol. 5, no. 1, pp. 8–9, 2020.

Uğur Alcan received his B.Sc. degree from Istanbul University in Electrical and Electronics Engineering 2019 and continuing his M.Sc in the same university. He is currently big data engineer in a tech company.

Firat Kacar received his B.Sc., M.Sc., and Ph.D. degrees from Istanbul University in Electrical and Electronics Engineering 1998, 2001, and 2005. He is currently Professor at the Electrical and Electronics Engineering Department of Istanbul University–Cerrahpasa. His current research interests include analog integrated circuits, active synthetic inductors, analog signal processing circuits, memristors and electronic device modeling. He is the author or co-author of about 150 papers published in scientific journals or conference proceedings. He is currently the chair of the Department of Electrical and Electronics Engineering (since 2018) at Istanbul University–Cerrahpasa.