Advances in Basic and Applied Sciences

journal homepage: https://abas.journals.ekb.eg/

Enhanced IoT Anomaly Detection Using Combined Machine and Deep Learning Techniques on the IoT-23 Dataset

Mariam Ahmed [1]*, E. Emary[2], Mahmoud A. AttaAlla[3]

[1] Faculty of computer Studies, Cybersecurity and forensics department, The Arab Open University (AOU), Cairo, Egypt.
[2] Program Coordinator at the Faculty of Computer Studies, Arab Open University (AOU), Assistant professor at faculty of Computers and Artificial Intelligence, Cairo University
3 Program Coordinator of MSc Cybersecurity and Forensics, Faculty of Computer Studies, The Arab Open University (AOU)

ARTICLE INFO

Article history:
Received 18 August 2025
Received in revised form 30 August 2025
Accepted 14 September 2025
Available online 3 December 2025

DOI :10.21608/abas.2025.415303.1078

Keywords: Internet of Things, Cyber security, Machine Learning, IoT-23 dataset, Anomaly Detection.

ABSTRACT

The Internet of Things (IoT) has revolutionized daily life and business operations by providing enhanced accuracy, efficiency, and speed. However, IoT systems face a growing risk of cyber threats, including malicious software and denial-of-service (DoS) attacks, which compromise the integrity and security of data. Despite the widespread adoption of IoT, many implementations lack robust cybersecurity measures, leaving them exposed to various threats. This highlights the urgent need to develop more resilient and effective defenses for IoT assets. This paper focuses on implementing anomaly detection for IoT systems utilizing the IoT-23 dataset, a comprehensive dataset containing both benign and malicious network traffic from IoT based devices. The study evaluates the performance of several machine learning (ML) and deep learning (DL) models, including decision trees, the Extreme Gradient Boosting (XGBoost) model, the Naïve Bayes model, and fully connected neural networks (FCNN).Our findings demonstrate that decision trees accomplished the highest accuracy of 98.9% while also requiring the shortest processing time, making them the most efficient model for anomaly detection in IoT systems. In contrast, the Naïve Bayes model performed the poorest, achieving an accuracy of only 50%. These results emphasize the significance of choosing the appropriate algorithm to enhance IoT security, ensuring effective anomaly detection and improved resilience against cyber threats.

[*] **Corresponding author E-mail:** *mariam_ahmed_mostafa@hotmail.com*

1. Introduction

The Internet of Things (IoT) represents a groundbreaking advancement in global information creation, following the enhancement of the Internet. IoT signifies a smart system enabling devices to connect, share information, and interact with one another via the internet. Through IoT, individuals can achieve objectives such as tracking, monitoring, managing, identifying, and locating with greater ease. The adoption of IoT has grown rapidly, becoming indispensable as internet usage and mobile devices have become integral parts of daily life. Furthermore, the number of IoT-enabled devices continues to rise steadily. However, the expansion of IoT has introduced significant security and privacy concerns for users. Since devices are interconnected through networks, attackers have access to a wider range of entry points for exploiting data. These connected devices frequently collect and transmit information, creating potential vulnerabilities. Many users lack an understanding of how IoT systems work, making it easier for attackers to steal sensitive data from individuals or their smart devices. information manufacturing following the advent of the Internet. The use of IoT has increased and has become essential since the utilization of the Internet and mobile technologies has become a necessity. Additionally, the number of IoT based devices is growing every day [1].

Nowadays, security has become the most challenging part for most prospective users of IoT systems. In response to these challenges, scientists have started exploring enhanced security measures over the past five years. Two primary approaches to IoT security have been identified: passive methods (like encryption and robust password protocols) and active methods. Among the emerging active techniques is the application of ML/DL, which are being utilized to identify and categorize cyberattacks effectively [2], [3]. From a technical perspective, these two technologies complement each other well. Machine learning algorithms require extensive datasets to develop accurate models. Large data sets can be provided by IoT systems, the Security Operations Center (SOC) staff in cyber security deal with enormous volumes of data that are gathered quickly from several sources. The analysts are frequently overloaded with information, which makes it impossible for them to respond and mitigate in a timely and sufficient manner. One well-known issue facing SOCs is called (data triage automation), where human performance is severely hampered by the domain’s intensity and incident detection [4] ,[5].

Moreover, the number of attacks and their motivations make the process of classifying and sorting them nearly impossible for the users. Examining the status of enterprise Cybersecurity is one of our objectives. Some surveys on companies in the cybersecurity space have been conducted, which assisted in understanding the current scenario. Results of such surveys have led to identify the type of assault and ascertain whether the system was secure[6]. By using filtering and prioritization techniques to handle information overload, such decision-support systems help organizations boost profits, improve customer satisfaction, and make quicker, more effective decisions in various sectors such as e-commerce, finance, and healthcare [7] .

We provide an overview of the different types of attacks across various levels of IoT infrastructure, along with the potential machine learning-based defenses against these attacks. These attacks are brought on by inadequate security data, poor-quality data, and learning algorithm performance, which may be crucial in supplying and enhancing IoT device security and privacy to identify and stop malware assaults on resource-constrained IoT based devices. In this study, we aim to compare different models based on their accuracy and time efficiency[8], [9].

2. Literature Review

Technology is changing day by day, so systems and applications are alike, but there are differences that make one better than the other [10], it is essential to acknowledge the role that prior practices and different systems play in understanding the ideas of completeness and perfection to properly evaluate their relevance. This comprehension facilitates the effort to keep a suitable pace while also addressing the extent of faults’ restrictions: flaws are removed by repeatable and predetermined methods, this method does not help us better understand current issues, but it also makes it easier to find possible areas where future procedures could be improved [7], in this section provides a quick overview of the different anomaly detection models and algorithms. IoT devices can be made safer and more private using a variety of methods. For example, [11], Doshi introduced a technique to identify DDoS attacks within the network layer that also has a low-cost ML approach, such as LSVM, KNN, Decision Tree, Random Forest, and NN. This approach was informed to accomplish high testing accuracy for these ML algorithms.

Additionally, in [12], symmetric keys are generated using a chaos-based encryption mechanism to ensure data integrity and authenticity during the transfer of data between the IoT device and the server. Moreover, in [13], the authors attempted to create an anomaly detection system by utilizing various classifiers, including random forest and k-NN, to construct the model. The system extracts relevant TCP packets from devices and analyzes key attributes such as labels, frame numbers, and length. The k-NN model categorizes the traffic into distinct classes, while the random forest model creates decision trees to identify potential attacks. In [14], authors tried to develop an anomaly detection system by applying various classifiers of Naïve Bayes, SVM, Decision Tree, and CNN to build the model and detect the attack according to its features.

In this study, we aim to compare different models based on their accuracy and time efficiency for detecting abnormalities in IoT based devices using the IoT-23 dataset [15], which includes both benign and malicious network traffic captured from various IoT based devices.

3. Methodology

3.1 proposed model

This paper proposed an IoT-based model designed to detect security anomalies effectively. In the suggested model, a traffic capture unit logs the data flow of network traffic from sensors to a processing unit, a central system, or a local/cloud-based computer, depending on the specific scenario. The compute unit stores the captured traffic in a database, allowing the model to be recalibrated for future use. Administrators can then select the most suitable model for anomaly detection based on performance and cost considerations. When anomalies are identified, the processing unit alerts the central system by sending messages or commands, which may involve actions such as removing questionable packets, checking for malicious software, conducting manual inspections, identifying Internet Protocol (IP) addresses, or alerting the user. Providing the most appropriate solution is essential, as each user operates under unique circumstances and may adopt different approaches to anomaly detection in IoT security [14].

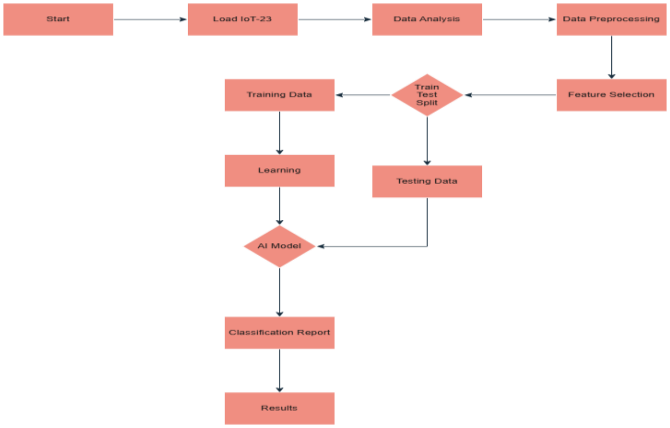

Figure 1: Enhanced IoT Anomaly Detection Using Combined Machine and Deep Learning Techniques on the IoT-23 Dataset

In addition, since the suggested model records the movement of network traffic and stores it in a database, a new dataset can be created and utilized in the future to recalibrate existing ML/DL algorithms, further enhancing their performance. The collected data is used to train ML algorithms such as the decision tree model, XGBoost, and Naïve Bayes, as well as deep learning techniques like fully connected neural networks (FCNN). Anomaly detection is then performed to identify irregularities in the system, as illustrated in Figure1.This detection process can be conducted either locally or in the cloud.

The dataset is split into separate subsets for training and testing, allowing conclusions to be drawn from the effectiveness of the trained algorithms. Based on the detection results, various actions can be implemented, such as Restricting the sender's IP address, discarding suspicious packets, notifying the user, or performing a manual inspection. If anomalies are detected, further checks can be performed to confirm potential attacks and ensure the system's security.

3.2 Dataset

The IoT-23 dataset was chosen from [15] . The dataset contains network traffic data from three distinct IoT based devices: Philips Hue, Somfy Door Lock, and Amazon Echo. Released in January 2020, it is a Extensive compilation of benign traffic and accurately labeled IoT malware incidents. Designed explicitly for developing ML and DL models, the dataset includes 23 features. Each of these features (derived from infected devices) corresponds to specific malware scenarios. The labels in the dataset include C&C, Attack, C&C-HeartBeat, C&C-FileDownload, C&C-HeartBeat-FileDownload, C&C-HeartBeatAttack, C&C-Torii, C&C-Mirai, DDoS, Okiru-Attack, Okiru, and PartOfAHorizontalPortScan.

3.3 Data Preprocessing

First, IoT-23 dataset [15] was loaded using the Pandas Python package. The data frames have 23 features. These features are (as shown in table 1): ts, uid, id.orig_h, id.orig_p, id.resp_h, id.resp_p, proto, service, duration, orig_bytes, resp_bytes, conn_state, local_orig, local_resp, missed_bytes, history, orig_pkts, orig_ip_bytes, resp_pkts ,resp_ip_bytes, tunnel_parents, label, and detailed-label.

Table 1.IoT-23 dataset's features

	First packet's time.
uid	Connection's unique identifier.
id	Endpoint ports / addresses of connections.
proto	Connection's transport layer protocol.
service	An application protocol's identification.
duration	The duration of the connection.
orig_bytes	The number of bytes that the sender sent.
resp_bytes	The number of bytes that the responder sent.

conn_state	Connection values.
local_orig	This will be true if the connection was made locally.
local_resp	This will be true if the connection is answered locally.
missed_bytes	Shows how many bytes are lost due to content gaps.
history	Stores the connections' state history as a string.
orig_pkts	The quantity of packets sent by the sender.
orig_ip_bytes	The quantity of IP-level bytes transmitted by the source.
resp_pkts	The quantity of packets sent by the respondent.
resp_ip_bytes	The quantity of IP-level bytes transmitted by the respondent.
tunnel_parents	uid values for any parent connections that are encapsulated.

1.Add/Drop features

a. Some features were discarded because their columns are empty, and others have very large null values.

b. The label and detailed label are combined in a new column called new label which contains ten different classes (as shown in Table 2): Malicious PartOfAHorizontalPortScan, Benign, MaliciousDDoS, Malicious C&C, Malicious Attack, Malicious C&C HeartBeat, Malicious C&C PartOfAHorizontalPortScan, Malicious C&C Torii, MaliciousC&C FileDownload, and Malicious FileDownload.

c. Moreover, new features are added such as packet_diff: which contains the difference between orig_pkts and resp_pkts, and ip_bytes_diff: which contains the difference between orig_bytes and resp_byteszaccording to analysis steps as shown in Figure2. So, features are: conn_state, id.resp_p, orig_pkts, history, new-label, orig_ip_bytes, pkts_diff, ip_bytes_diff.

Table 2. Counts of label Types

	Count
Malicious PartOfAHorizontalPortScan	10427048
Benign	8755042
Malicious DDoS	5778154
Malicious C&C	15039
Malicious Attack	8717
Malicious C&C HeartBeat	940
Malicious C&C PartOfAHorizontalPortScan	888
Malicious C&C Torii	30
Malicious C&C FileDownload	26
Malicious FileDownload	3
Name: count, dtype: int64	Name: count, dtype: int64

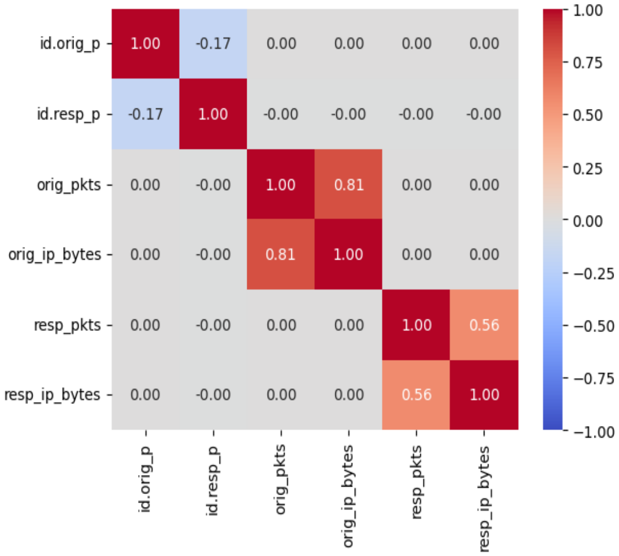

Figure 2: Correlation matrix describes the relation between features.

2. Handling missing values

All the rows which contained missed values were discarded.

3. Data splitting

The data set was split into a testing data set (size of 0.2) and a training data set (size of 0.8) for validation.

4. performance evaluation and analysis

A. System and Environment Configuration

The experiments were conducted online on Kaggle using a T4 GPU. Python version 3.10.12 and PyTorch environments were utilized for the experiments.

B. Metrics Evaluation

Several metrics are employed to evaluate the efficiency of the model, which mainly depend on precision, support, recall, and F1 score [16]:

1. *Time*

Consideration is given to how long a model takes to execute a specific ML/DL algorithm. As mentioned earlier, an algorithm with a lengthy runtime may not be appropriate for IoT technology.

2. True Positives (TP)

The result when the positive class is accurately predicted by the model.

3. False Positives (FP)

The result when the positive class is predicted by the model wrongly.

4. Precision

It is defined as a metric for computing the accurately recognized positives in an algorithm [17] and is provided by:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (1)$$

5. Recall

It is a measurement of the real quantity to appropriate positives that are recognized [18]and provided by:

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2)$$

6. F1 score

This metric calculates the weighted average of precision and recall, accounting for both false positives and false negatives. It is considered a more comprehensive and reliable measure [18]. It is

provided by: $\text{F1} = 2* \frac{Precision \times Recall}{Precision + Recall}$ (3)

7. Support score

This is a performance metric that represents how often each label is correctly identified as true.

C. Test Results for Machine Learning and Deep Learning Algorithms

1. Extreme Gradient Boosting (XgBoost)

XGBoost is a scalable gradient boosting framework created to enable scalable and efficient training of machine learning models. It utilizes an ensemble learning approach, combining the outputs of several weaker models to produce more accurate and reliable predictions. XGBoost has gained significant popularity and widespread adoption due to its ability to process large datasets efficiently while delivering excellent performance in both regression and classification tasks[19]. As illustrated in Table 3, the XGBoost algorithm achieves an accuracy of 98.8%, and the time required for execution is 17.2 seconds.

2. Decision Tree.

A decision tree is a supervised learning method that does not rely on parameter assumptions used for solving both regression and classification problems. It is structured as a hierarchical model consisting of a root node, branches, internal nodes, and leaf nodes, which represent different decision points and outcomes. [20]. As illustrated in Table 3, the Decision Tree algorithm achieved an accuracy of 98.9%, and the time required for execution is 0.47 seconds. Among our results, the Decision Tree demonstrated the highest accuracy.

3. Naive Bayes Model.

Naïve Bayes is a supervised machine learning algorithm designed for classification tasks, such as text categorization. It operates by applying probabilistic principles and belongs to the family of generative learning algorithms.[21]. Naïve Bayes focuses on modeling the input distribution for a specific class or category. Unlike discriminative classifiers such as logistic regression, it does not determine which features are most significant for distinguishing between classes[22]. As illustrated in Table 3, the Naïve Bayes model achieved an accuracy of 50%, and the time required for execution is 2.32 seconds. It recorded the lowest accuracy among our results.

4. Fully Connected Neural Network (FCNN).

Neural Networks is a machine learning model that uses procedures that resemble biological neurons. They cooperate to recognize occurrences, evaluate possibilities, and reach conclusions to make decisions that are comparable to those made by the human brain [23]. As illustrated in Table 3, the accuracy for the FCNN algorithm is 98.6 %, and the time required for execution is 70 seconds.

D. Results Comparison

In Table 3, it displays the examination results of the used ML models, the decision tree comes with the best result, which has an accuracy of 98.9% and takes 0.47 seconds, while XGBoost comes with an accuracy of 98.8% but takes 17.2 seconds, that makes it less effective than the decision tree model. Neural networks come with an accuracy of 98.6% and take 70 seconds, which is very slow for more work. Lastly, Naïve Bayes is the lowest accuracy that is 50% and taking 2.32 seconds, which is not effective in many tasks that need high performance. As shown in Table 4, our decision tree model comes with superior results compared to paper [14] , which achieved a higher accuracy of 98.9% compared to 73%. Additionally, our Naïve Bayes model comes with a higher accuracy of 50% than the result in paper [14] of 30%, and it takes a slightly shorter time 2.32 seconds vs. 6 seconds. So, our models come with high accuracy with only a moderate speed in predication time that makes them more effective for applications[14].

Table 3. Our results of different models on IoT-23 dataset

Method	Accuracy	Precision	Recall	F1-score	Time Cost
XGBoost	98.8%	99%	99%	99%	17.2 s
Decision Tree	98.9%	99%	99%	99%	0.47 s
Naïve Bayes	50%	100%	50%	56%	2.32 s
FCNN	98.6%	99%	99%	99%	70 s

Table 4. Results Comparison with paper[14] on IoT-23dataset

Method	Testing Accuracy	Time Cost
Decision Tree(ours)	98.9 %	0.47s
Decision Tree (paper [14])	73%	3 s
Naïve Bayes(ours)	50%	2.32s
Naïve Bayes(paper[14])	30%	6 s

Conclusion and Future Scope

This paper presents an anomaly detection system designed to enhance IoT security by evaluating the performance of various ML/DL algorithms and methodologies. Based on our findings, decision trees proved to be the most accurate algorithm, achieving an impressive accuracy of 98.9% while also requiring the shortest execution time among all the machine learning methods tested. This result provides valuable insights into the effectiveness, efficiency, and comparability of different approaches. On the other hand, the Naïve Bayes model exhibited the lowest performance, with an accuracy of just 50%, highlighting its limitations in this context.

Although our model demonstrated encouraging accuracy when tested on the IoT-23 dataset, two major challenges remain: generalizability and interpretability. These obstacles must be addressed to unlock the full potential of the anomaly detection system. Generalizability refers to the model's ability to perform consistently well across diverse datasets and real-world scenarios, while interpretability involves making the model's decision-making process more transparent and understandable to users.

To further enhance the system, future studies should focus on testing the model across diverse datasets and adversarial attack conditions to ensure robustness and adaptability in real-world applications. Additionally, exploring hybrid approaches that combine multiple algorithms or domain-specific optimizations may offer pathways to overcome these limitations. By addressing these challenges, we can move closer to developing a refined model capable of significantly improving DDoS detection and strengthening network security in IoT environments. Achieving this would represent a major milestone in the field of IoT anomaly detection and pave the way for more secure and resilient IoT systems in the future.

Ethics approval

Not applicable.

Availability of data and material

Not applicable.

Conflict of interest

The author declared no potential conflicts of interest concerning this article’s research, authorship, and publication.

Funding

The author received no financial support for the research.

Acknowledgment

The author thanks anonymous reviewers for their comments that enhance the paper’s present form.

References

[1] “Securing Smart Cities: A Cybersecurity Perspective on Integrating IoT, AI, and Machine Learning for Digital Twin Creation,” 2024.

[2] Srikanth Reddy Vutukuru 2 Srinivasa Chakravarthi Lade SecureIoT: Novel Machine Learning Algorithms for Detecting and Preventing Attacks on IoT Devices,”

2023.

[3] Vinay Tila Patil 2 Shailesh Shivaji Deore IoT-Guardian: Advanced Detection of DDoS Attacks in IoT Systems Using CNNs," 2024.

[4] M. Majid and K. Ariffi, "Success Factors for Cyber Security Operation Center (SOC) Establishment," European Alliance for Innovation n.o., Oct. 2019. doi: 10.4108/eai.18-7-2019.2287841.

[5] H. Kayan, M. Nunes, O. Rana, P. Burnap, and C. Perera, "Cybersecurity of Industrial Cyber-Physical Systems: A Review," Jan. 2021, [Online]. Available: http://arxiv.org/abs/2101.03564

[6] N. H. A. Rahim, S. Hamid, L. M. Kiah, S. Shamshirband, and S. Furnell, "A systematic review of approaches to assessing cybersecurity awareness," *Kybernetes*, vol. 44, no. 4, pp. 606–622, Apr. 2015, doi: 10.1108/K-12-2014-0283.

[7] Pawlicka, M. Pawlicki, R. Kozik, and R. S. Choraś, "A systematic review of recommender systems and their applications in cybersecurity," Aug. 01, 2021, *MDPI AG*. doi: 10.3390/s21155248.

[8] "Securing Smart Cities: A Cybersecurity Perspective on Integrating IoT, AI, and Machine Learning for Digital Twin Creation," 2024.

[9] N.-A. Stoian, "Machine Learning for Anomaly Detection in IoT networks: Malware analysis on the IoT-23 Data set."

[10] N. N. Ahmed and K. Nanath, "Exploring Cybersecurity Ecosystem in the Middle East: Towards an SME Recommender System," *Journal of Cyber Security and Mobility*, vol. 10, no. 3, pp. 511–536, 2021, doi: 10.13052/jcsm2245-1439.1032.

[11] R. Doshi, N. Apthorpe, and N. Feamster, "Machine Learning DDoS Detection for Consumer Internet of Things Devices," Apr. 2018, doi: 10.1109/SPW.2018.00013.

[12] T. Song, R. Li, B. Mei, J. Yu, X. Xing, and X. Cheng, "A Privacy Preserving Communication Protocol for IoT Applications in Smart Homes," *IEEE Internet Things J*, vol. 4, no. 6, pp. 1844–1852, Dec. 2017, doi: 10.1109/JIOT.2017.2707489.

[13] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, "IoT Security Techniques Based on Machine Learning: How Do IoT Devices Use AI to Enhance Security?," *IEEE Signal Process Mag*, vol. 35, no. 5, pp. 41–49, Sep. 2018, doi: 10.1109/MSP.2018.2825478.

[14] Y. Liang and N. Vankayalapati, "Machine Learning and Deep Learning Methods for Better Anomaly Detection in IoT-23 Dataset Cybersecurity."

[15] "IoT-23 Dataset "httpswww.stratosphereips.orgdatasets-iot23"".

[16] Ž. Vujović, "Classification Model Evaluation Metrics," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, pp. 599–606, 2021, doi: 10.14569/IJACSA.2021.0120670.

[17] B. Juba and H. S. Le, "Precision-Recall versus Accuracy and the Role of Large Data Sets." [Online]. Available: www.aaai.org

[18] R. Yacouby Amazon Alexa and D. Axman Amazon Alexa, "Probabilistic Extension of Precision, Recall, and F1 Score for More Thorough Evaluation of Classification Models."

[19] R. Ravi* and Dr. B. Baranidharan, "Crop Yield Prediction using XG Boost Algorithm," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 5, pp. 3516–3520, Jan. 2020, doi: 10.35940/ijrte.D9547.018520.

[20] B. Charbuty and A. Abdulazeez, "Classification Based on Decision Tree Algorithm for Machine Learning," *Journal of Applied Science and Technology Trends*, vol. 2, no. 01, pp. 20–28, Mar. 2021, doi: 10.38094/jastt20165.

[21] N. B. Classifier, "Data Mining Lecture # 7 Naïve Bayes Classifier."

[22] S. Taheri and M. Mammadov, "Learning the naive bayes classifier with optimization models," *International Journal of Applied Mathematics and Computer Science*, vol. 23, no. 4, pp. 787–795, 2013, doi: 10.2478/amcs-2013-0059.

[23] L. N. N. Do, N. Taherifar, and H. L. Vu, "Survey of neural network-based models for short-term traffic state prediction," Jan. 01, 2019, *Wiley-Blackwell*. doi: 10.1002/widm.1285.