ENSEMBLE MACHINE LEARNING APPROACH FOR IOT INTRUSION DETECTION SYSTEMS

Baseem A. Kadheem Hammood [1]

[1]*Department of Computer Science*
Informatics Institute for Postgraduate Studies, Iraqi Commission for Computers and Informatics
Baghdad, Iraq
ms202130656@iips.icci.edu.iq

Ahmed T. Sadiq[2]

[2]*Department of Computer Science*
University of Technology

Baghdad, Iraq
Ahmed.T.Sadiq@uotechnology.rdu.iq

Abstract **- The rapid growth and development of the Internet of Things (IoT) have had an important impact on various industries, including smart cities, the medical profession, autos, and logistics tracking. However, with the benefits of the IoT come security concerns that are becoming increasingly prevalent. This issue is being addressed by developing intelligent network intrusion detection systems (NIDS) using machine learning (ML) techniques to detect constantly changing network threats and patterns. Ensemble ML represents the recent direction in the ML field. This research proposes a new anomaly-based solution for IoT networks utilizing ensemble ML algorithms, including logistic regression, naive Bayes, decision trees, extra trees, random forests, and gradient boosting. The algorithms were tested on three different intrusion detection datasets. The ensemble ML method achieved an accuracy of 98.52% when applied to the UNSW-NB15 dataset, 88.41% on the IoTID20 dataset, and 91.03% on the BoTNeTIoT-L01-v2 dataset.**

Index Terms - intrusion detection system, Machine Learning, IoT, Ensemble.

I. INTRODUCTION

Discovering emerging and unknown attacks requires an approach that can detect Internet of Things (IoT) intrusion; machine learning (ML) possesses this ability [1]. The rapid growth of cyberattacks has resulted in the need of IoT's security architecture for intrusion detection. The security field faces serious challenges in the development of technology and the IoT. Current security methods do not provide adequate protection; hence, cyberattacks are increasing. [2].

With the use of an ML-based approach, an intrusion detection system (IDS) was proposed for use on the IoT. The proposed model can be trained on different sources from large and classified datasets. This model can work effectively after being trained on smaller-sized data and classifying them in the target domain [3].

Another IoT IDS has been proposed using ML and enhanced transient search optimization. The proposed system uses an enhanced transient search optimization algorithm to optimize the hyperparameters of the ML model. The outcomes of this paper show that the recommended system outperforms other IDS in terms of accuracy and false alarm rate [4].

This work uses ensemble ML methods to detect intrusion in IoT networks. This article is organized as follows: Section 2 presents the related work, Section 3 presents the IoT intrusion detection system, Section 4 introduces ensemble ML, Section 5 provides the classifiers, Section 6 presents the proposed method, Sections 7 and 8 detail the experimental results, and Section 9 concludes this paper.

II. RELATED WORK

In this section, some previous works in the field of IoT IDS are reviewed.

In [5], feature sets were used, and ML methods using multiple over-cluster approaches (artificial neural networks (NN), backing machines, and random forests (RF), and message queue telemetry transport (MQTT), a transport metric for waiting messages, UNSW-NB15, which is feature-based by TCP. The best features in the two groups were obtained, with high accuracy and less time for the ML algorithms. RF, binary, and the use of radio frequency on stream data and MQTT achieved accuracies of 97.37%, 98.67%, and 97.54%, respectively.

In [6], four algorithms—naive Bayes (NB), RF, J48, and zero—were utilized to categorize cyberattacks on the UNSW-NB15 dataset. Two groups were created using the UNSW-NB15 dataset using K-means and expectation maximization clustering techniques, depending on whether the objective attack is used or regular network traffic only. Following the classification above to create a subset of features, correlation-based features were used. The techniques are useful for research on intrusion detection in widespread networks. The results demonstrate that the RF and J48 algorithms achieved accuracies of 97.59% and 93.78%, respectively.

In [7], NN, logistic regression (LR), NB, decision tree (DT), SGD, and RF classifiers were evaluated empirically and tested using the UNSW-NB15 dataset. Accuracy indicates a correlation between classifiers. The RF classifier outperformed the other methods, having an accuracy of 95.43%.

In [8], the proposed system called MidSiot is used on the IoT. It consists of several stages, including identifying and classifying attacks and real network traffic, and achieved an average accuracy of 99.68%.

In [9], an IDS called Pearson correlation coefficient–convolutional neural networks (PCC-CNN) was established

for the deep learning model. Intrusion detection was performed by collecting features, detecting changes, and extracting linear operations. Attacks are detected using the binary classifier based on three sets of data, achieving 98%, 99%, and 98% similarity accuracy in the three datasets.

In [10], a modified IDS was proposed based on ML, and the RF algorithm was used to enter features. The output of the IoTID20 dataset after removing the nominal features is 79 characters. The accuracy of the proposed model was 96.5%. The categorical values were converted into numeric values because the inputs of all algorithms must be numeric values. Most researchers used binary classification. In this paper, multiple classification of 9 or 10 categories will be used.

III. IoT INTRUSION DETECTION SYSTEM

The intrusion detection process involves monitoring and analyzing the events in a computer system or network for indicators of intrusions (attempts to undermine the confidentiality, integrity, or availability of a computer system or network). Attackers who access systems over the Internet, authorized users who try to gain unauthorized access rights, and authorized users who abuse their powers are all sources of intrusion. This monitoring and analysis process is automated by software or hardware solutions.

Intrusion detection enables organizations to defend their systems against risks brought on by growing network connections and dependence on information systems. Security professionals should decide whether to utilize intrusion detection rather than decide which intrusion detection features and capabilities to deploy, given the severity and type of contemporary network security threats. IDSs are now widely recognized as crucial to any organization's security architecture. Even though IDSs have been shown to improve system security, many organizations still need justification to purchase an IDS [11].

A security system for an IoT environment needs to be created while considering security precautions. Data-oriented security mechanisms must be prioritized to stop hostile users from gaining unauthorized access to data sources. Focusing on data integrity and confidentiality is crucial because doing so significantly lowers the major security dangers in an IoT context. Conventional security procedures, which are designed using cryptographic techniques, are not often used in IoT environments because of the huge amount of data. Network problems will be lessened if threats are discovered quickly. Conventional security models take more time to evaluate such a large volume of data to identify the risks. A bad user just needs brief unauthorized access to data to obtain sensitive information, and changing that information might significantly negatively affect the user. By blocking access from unauthorized users, an IDS identifies intruders and safeguards the network and data. A central IDS that monitors the network and distant nodes and detects intrusions might be employed to decrease this complexity. As a result, the network administrator receives a notification to take action on the security vulnerabilities [12].

Three steps make up the IDS's functionality. The first monitoring phase is based on network or host sensors. The second phase is analysis, which involves feature extraction and pattern recognition. The last stage is detection, which involves finding network anomalies or intrusions. IDS aids in quickly detecting vulnerabilities and monitoring and analyzing data, services, and networks as well as traffic analysis via efficient network management. It enhances data, network secrecy, and integrity while defending the network against threats. An IDS compiles and examines the system's data stream to find any malicious or dangerous activity. Traditional IDS design lacks real-time security for huge volume data streams and primarily focuses on providing security for Internet management features.

The IDS operates primarily in the network layer of the IOT system [11]. The network layer of an IoT NIDS monitors Internet data transferred between the network's devices. Also, it serves as a second line of defense to detect and protect the network from threats from unauthorized users [12].

Typically, an IDS consists of sensors, which collect the data to be analyzed by IDS tools. These tools report abnormal activities such as attacks or unauthorized access. An intrusion can be defined as any assault that compromises the availability, confidentiality, or integrity of information. An IoT system's IDS should be able to analyze data packets and respond in real time at different IoT network levels utilizing different protocol stacks and adjust to different threats [13].

IV. ENSEMBLE MACHINE LEARNING

Ensemble approaches may combine many algorithms instead of just one ML classification algorithm. The model's accuracy is enhanced by using this method. Algorithms for supervised learning are ensemble approaches. Different training algorithms benefit from ensemble approaches, which increase the training accuracy to raise the testing accuracy. The ensemble approach may use different training algorithms to provide flexible training [14].

V. CLASSIFIERS

ML is a subtype of artificial intelligence that allows a computer to make decisions independently without human input, enabling computers to learn independently without being explicitly programmed. The fundamental objective of ML is to create computer software that can access data and use it for learning procedures.

Several kinds of ML exist [15]. Six ML methods (both linear and nonlinear) were extensively utilized for IDS data classification. Therefore, the background of the ensemble ML

and six methods (DT, GB, and extra tree) should be understood so they can be utilized for intrusion detection.

A. Decision Tree

The DT is a supervised learning technique that is used to handle classification and regression problems and is most often selected to do both. It is a tree-structured classifier in which each leaf node represents the classification structure, and the interior nodes reflect the dataset's characteristics. A DT comprises two nodes: the decision node and the leaf node. In contrast to leaf nodes, which indicate choices' results and have no other branches, decision nodes are used to make decisions and contain multiple branches. Two possible answers represent each question in a DT: "yes" or "no," which enables the creation of branches. The tree could be split up into smaller trees (Figure 1) [16].

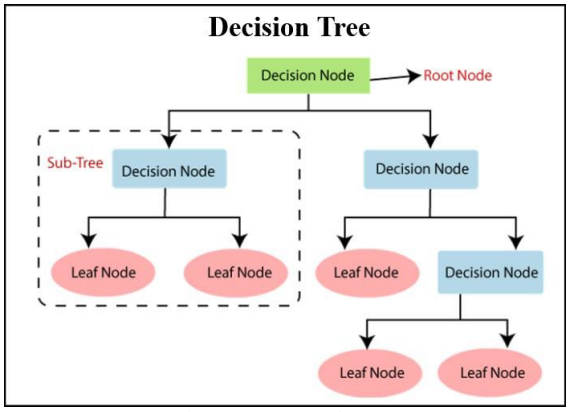

Fig. 1 Schematic of a DT.

B. Random Forest

Many DT classifiers, each built using a random vector sampled independently from the input vector, make up the RF classifier. Each tree casts a unit vote for the dominant class to classify an input vector. Most DTs simulate scenarios that do not operate well but may provide the foundation for other trees to work better. The Gini index, which measures an attribute's impurity in classes, is used as an attribute selection metric. Every time a tree is developed to its maximum depth, a mix of features and fresh training data is utilized. These mature trees have yet to be trimmed. This ability is one of the RF classifier's main benefits over other DT approaches (Figure 2) [17][18].

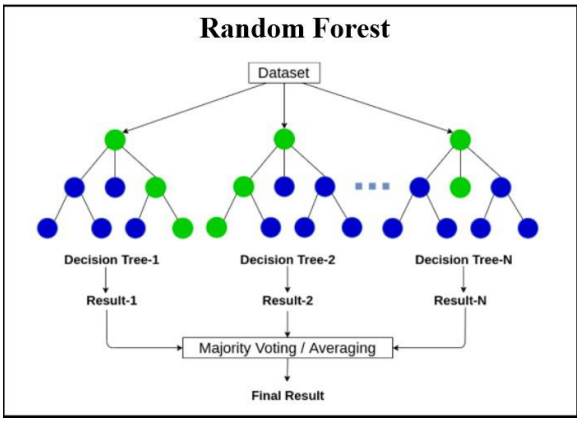

Fig. 2 Random forest.

C. Naive Bayes

The Bayes theorem is the foundation of NB classifiers. It is based on conditional probability, which refers to the chance that an event (A) will occur given that another event (B) has already occurred. Essentially, the theorem permits a hypothesis to be revised whenever new data are presented. It is a simple and effective predictive modeling technique. The model may directly extract two types of probabilities from the training data: the likelihood of each class and the conditional probability for each class given each x value. The Bayes theorem may be used to forecast new data using the probability model, as shown in Eq. (2) [19].

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (1)$$

D. Logistic Regression

LR is used to predict a binary result (1 or 0, yes or no, true or false) given a collection of independent factors to depict binary or categorical outcomes. When the log of chances is used as the dependent variable when the outcome variable is categorical, LR is a particular instance of linear regression (Figure 3) [20], [18], [21].

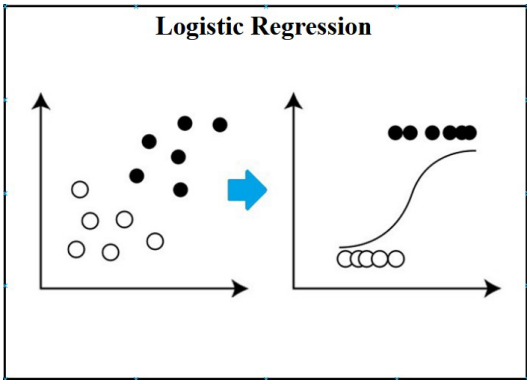

Fig. 3 Logistic regression.

E. Gradient Boosting

Gradient-boosted machines (GBMs) are popular ML algorithms that are widely used in many different sectors and are one of the most effective ways to win Kaggle tournaments. While RF constructs an ensemble of deep, autonomous trees, GBMs construct an ensemble of shallow, weak, consecutive trees, with each tree learning from and improving upon the previous ones. These numerous weak consecutive trees come together to form a potent "committee," frequently challenging other algorithms [22].

F. Extra Tree

The different trees and RF differ primarily in two ways. First, unlike RF, the different trees do not create the training subset for each tree using the tree bagging step. All DTs in the ensemble are trained using the whole training set. Second, the extra trees randomly choose the best characteristic and its corresponding value during the node-splitting stage. As a result of these two variations, the trees are less prone to overfitting and have improved performance [23].

VI. PROPOSED METHOD

This research used three datasets: UNSW-NB15, IoTID-20, and BotNetIoT. Six types of ML architectures were tested to determine the effectiveness of various ML architectures on these datasets. Before the models were trained on the datasets, the data underwent preparation. Subsequently, two of the datasets, namely, UNSW-NB15 and BotNetIoT, were split into training and testing sets in a 70:30 ratio, while the IoTID-20 dataset was split into training and testing sets in an 80:20 ratio. The training data were then fed into ML algorithms, which included LR, NB, DT, extra trees, RF, and gradient boosting. Finally, the strongest results were voted on by using the ensemble method. The effectiveness of the trained models was evaluated using the test data, as presented in Figure 4.

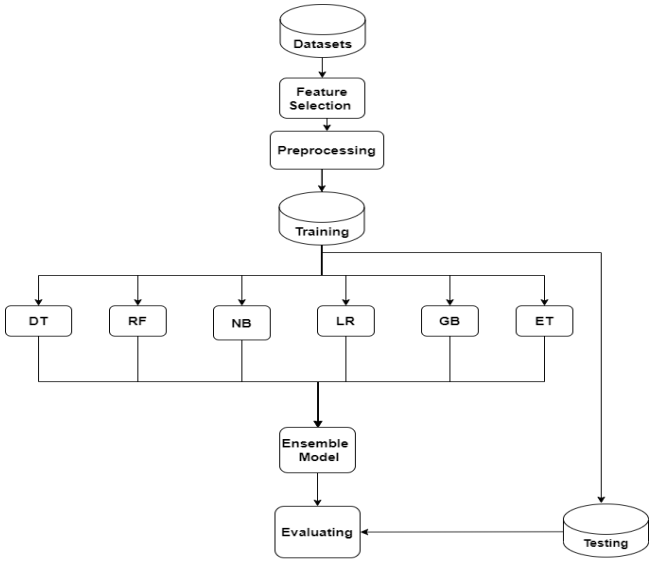

Fig. 4 Applying ensemble ML algorithms to different datasets.

A. Datasets

This paper used three IoT intrusion detection datasets. First, the UNSW-NB15 [24] dataset is a labeled network traffic dataset that contains more than two million records of network traffic captured from a realistic network environment, including benign and malicious attributes. The dataset includes 49 network features extracted from each n flow and labels that indicate whether the traffic is malicious or benign, making it a useful resource for evaluating the effectiveness of intrusion detection methods for IoT networks. Second, the IoTID-20 [25], [26] dataset is a publicly available labeled dataset that was specifically designed for IoT intrusion detection research. It contains network traffic data collected from a real-world IoT environment with 20 different types of IoT devices. The dataset includes benign and malicious attributes, with a total of 15 attack scenarios generated by using various network attacks, such as brute-force attacks, DoS attacks, and malware infections. The IoTID-20 dataset is useful for evaluating the effectiveness of various IDS and ML algorithms in detecting IoT-specific attacks. Table I shows the attack types in each dataset.

TABLE I TYPES OF ATTACKS IN EACH DATASET.

Dataset	Attacks
IoTID20	Mirai-Ackflooding; DoS-Synflooding; Scan Port OS; Mirai-Hostbruteforceg; Mirai-UDP Flooding; Mirai-HTTP Flooding; Scan Hostport; MITM ARP Spoofing
UNSW_NB15	reconnaissance; shellcode; exploit; fuzzes; worms; denial-of-service attacks; backdoors; analysis; generic
BoTNeTIoT	combo; junk; scan; tcp; udp; ack; syn ; udpplain

This study uses an IoT dataset for IDS, specifically the Malicious BotNet dataset (BotNetIoT), which consists of data files collected during the detection of IoT botnet attacks on a cybersecurity system. This dataset is publicly available on Kaggle [27].

To create this dataset, researchers used Wireshark software to capture network traffic data from nine IoT devices in a local network. The data were collected in packet capture (PCAP) file format, which is commonly used for network analysis. The PCAP file contains data packets from the network, including 23 statistical features for the central switch in the network.

The data in the BotNetIoT dataset include benign and malicious traffic, with the malicious traffic generated by various IoT-specific attacks, such as botnets and infiltration attacks. The dataset is useful for evaluating the effectiveness of IDS in detecting IoT-specific attacks and assessing network health. It is also useful for training and testing ML algorithms for IoT intrusion detection. Table II shows the specification of the three datasets.

TABLE II SPECIFICATION OF DATASETS

Dataset	No. of objects	No. of features	No. of classes
IoTID20	625783	86	9
UNSW_NB15	2540047	49	10
BoTNeTIoT-L01-v2	7062606	27	9

B. Data Preprocessing

1) Data Cleaning: In this preprocessing step, the features that were not useful in the prediction process and had only one value are deleted. Moreover, rows that contain duplicate data were identified and deleted.

2) Handling Missing Values: The dataset has some missing values, which were substituted with the value of 0.

3) Normalization: Feature normalization is an essential step in data preprocessing. Data normalization is a practical approach to improving ML accuracy. The standard scaler transforms the data of the three datasets to a range between 0 and 1. It was implemented before being integrated into the proposed deep learning classification model, as shown in Eq. (2)

$$x' = \frac{x - (x)}{(x) - min(x)} \quad (2)$$

C. Ensemble Machine Learning Approach to Detecting IoT Intrusion

A voting-based ensemble classification technique is used. Several voting procedures exist, such as hard voting (voting based on a majority) and soft voting. Soft voting may be performed by using the average of probabilities, the product of probabilities, the lowest or maximum of probabilities, or none of them.

In this work, hard voting (voting based on a majority) was used to assess the voting mechanisms.

VII. EXPERIMENTAL RESULTS

In this part, the confusion matrix-based findings for multi-class classification were provided. The model's performance based on accuracy, precision, recall, and F1 score was assessed. In contrast to recall, which is determined by dividing the total number of positive class values into the test data by the number of true positive predictions, precision is calculated by dividing the total number of true positive predictions by the total number of positive class values predicted. The weighted average of recall and accuracy is the F1 score. Accuracy is determined by dividing the total number of forecasts by the number of right predictions (including true positive and true negative predictions). Poor recall is reflected by a large number of incorrect negative predictions, and low accuracy is indicated by a high proportion of false positive predictions. A high F1 score indicates accuracy and recall that are in balance, with few false negatives and positives. These measures were calculated using the appropriate equations, which were based on sources [28-31].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

$$F1\ Score = 2 \times \frac{Precision*Recall}{Precision + Recall} \quad (6)$$

where TP is the true positive, TN is the true negative, FP is the false positive, and FN is the false negative.

The experiments were conducted using the Google Colab platform, which includes an NVIDIA Tesla T4 GPU, 12 GB of RAM, a 53 GB hard disk, and a CPU frequency of 2.30 GHz.

TABLE III PERFORMANCE METRICS IN THE IOTID20 DATASET

Algorithm	Accuracy	Recall	Precision	F1Score
Logistic regression	74.59%	56.56%	56.45%	55.83%
Naïve Bayes	47.01%	46.19%	44.33%	33.47%
Decision tree	83.91%	77.14%	78.12%	77.56%
Extra tree	83.39%	75.98%	77.24%	76.55%
Random forest	83.70%	76.51%	76.64%	76.58%
Gradient boosting	**88.41%**	81.90%	82.87%	82.00%

Table III shows the preference for gradient boosting algorithms over other algorithms, and the accuracy of this algorithm was 88.41%.

TABLE IV PERFORMANCE METRICS IN THE UNSW_NB15 DATASET.

Algorithm	Accuracy	Recall	Precision	F1 Score
Logistic regression	96.36%	26.38%	54.43%	28.80%
Naïve Bayes	86.86%	39.22%	19.26%	19.97%
Decision tree	98.30%	53.86%	56.57%	54.79%
Extra tree	98.41%	47.97%	57.93%	50.28%
Random forest	98.51%	51.46%	56.49%	52.15%
Gradient boosting	**98.52%**	98.43%	98.53%	98.52%

Table IV shows the preference for gradient boosting algorithms over other algorithms, and the accuracy of this algorithm was 98.52%.

TABLE V PERFORMANCE METRICS IN THE BOTNETIOT DATASET.

Algorithm	Accuracy	Recall	Precision	F1 Score
Logistic regression	75.06%	73.98%	69.77%	70.31%
Naïve bayes	56.17%	63.70%	60.38%	56.10%
Decision tree	91.02%	88.88%	96.84%	87.42%
Extra tree	**91.03%**	88.90%	97.29%	87.43%
Random forest	90.99%	88.85%	97.11%	87.36%
Gradient boosting	90.54%	87.81%	96.21%	86.44%

Table V shows the preference for extra tree algorithms over other algorithms, and the accuracy of this algorithm was 91.03%.

When the ensemble classification method was applied to the IoTID20 dataset, it reached an accuracy of 88.41%, an accuracy of 98.52% on the UNSW_NB15 dataset, and 91.03% on the BoTNeTIoT fataset.

In this study, this method was compared with methods in several recent studies. Table VI provides a comparison of the overall performance in multiple classifications on the UNSW_NB15 dataset in terms of accuracy. Table VII compares studies conducted on the IoTID20 dataset for subcategories in terms of accuracy. The proposed approach outperformed the other methods in terms of accuracy measures.

TABLE VI GENERAL COMPARISON OF MULTIPLE CLASSIFICATION ACCURACY MEASURES FOR THE UNSW_NB15 DATASET.

Method	Accuracy
Ref. [5]	97.54%
Ref. [6]	97.59%
Ref. [7]	95.43%
Proposed method (ensemble method)	**98.52%**

TABLE VII GENERAL COMPARISON OF SUBCATEGORIES WITH PRECISION MEASURES OF THE IOTID20 DATASET.

Method	Accuracy
Ref. [10]	83.7%
Proposed method (ensemble method)	**88.41%**

VIII. CONCLUSION

Ensemble techniques mix several learning algorithms to achieve prediction performance that is better than that of any one of the component learning algorithms alone. Empirically, ensemble ML provides more accurate findings when models exhibit considerable variations. As a result, many ensemble approaches encourage variation among the models they combine. In this research, three intrusion detection datasets for the IoT (IoTID20, UNSW-NB15, and BoTNeTIoT-L01-v2) were employed to evaluate the performance of the ensemble classification method. The results indicate a preference for the ensemble classification method over the other algorithms, with accuracy rates of 88.41% on the IoTID20 dataset, 98.52% on the UNSW-NB15 dataset, and 91.03% on the BoTNeTIoT-L01-v2 dataset. In conclusion, ML approaches show great potential for IoT IDS. They can provide important solutions with their anomaly-based approach and ability to detect unknown attacks. As a future research direction, a recommendation using several feature selection methods can be formulated. Hybrid feature selection methods can also be used.

REFERENCES

[1] *S. Tsimenidis, T. Lagkas, and K. Rantos, "Deep Learning in IoT Intrusion Detection," Journal of Network and Systems Management, vol. 30, no. 1, Jan. 2022, doi: 10.1007/s10922-021-09621-9.*

[2] *A. Fatani, M. A. Elaziz, A. Dahou, M. A. A. Al-Qaness, and S. Lu, "IoT Intrusion Detection System Using Deep Learning and Enhanced Transient Search Optimization," IEEE Access, vol. 9, pp. 123448–123464, 2021, doi: 10.1109/ACCESS.2021.3109081.*

[3] *Sk. T. Mehedi, A. Anwar, Z. Rahman, K. Ahmed, and R. Islam, "Dependable Intrusion Detection System for IoT: A Deep Transfer Learning-based Approach," Apr. 2022, doi: 10.1109/TII.2022.3164770.*

[4] *A. Awajan, "A Novel Deep Learning-Based Intrusion Detection System for IoT Networks," Computers, vol. 12, no. 2, Feb. 2023, doi: 10.3390/computers12020034.*

[5] *M. Ahmad, Q. Riaz, M. Zeeshan, H. Tahir, S. A. Haider, and M. S. Khan, "Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-set," EURASIP J Wirel Commun Netw, vol. 2021, no. 1, Dec. 2021, doi: 10.1186/s13638-021-01893-8.*

[6] *M. Hammad, W. El-Medany, and Y. Ismail, "Intrusion Detection System using Feature Selection with Clustering and Classification Machine Learning Algorithms on the UNSW-NB15 dataset," in 2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies, 3ICT 2020, Institute of Electrical and Electronics Engineers Inc., Dec. 2020. doi: 10.1109/3ICT51146.2020.9312002.*

[7] *G. Kocher and G. Kumar, "Performance Analysis of Machine Learning Classifiers for Intrusion Detection using UNSW-NB15 Dataset," Academy and Industry Research Collaboration Center (AIRCC), Dec. 2020, pp. 31–40. doi: 10.5121/csit.2020.102004.*

[8] *N. Dat-Thinh, H. Xuan-Ninh, and L. Kim-Hung, "MidSiot: A Multistage Intrusion Detection System for Internet of Things," Wirel Commun Mob Comput, vol. 2022, 2022, doi: 10.1155/2022/9173291.*

[9] *M. Bhavsar, K. Roy, J. Kelly, and O. Olusola, "Anomaly-based intrusion detection system for IoT application," Discover Internet of Things, vol. 3, no. 1, p. 5, May 2023, doi: 10.1007/s43926-023-00034-5.*

[10] *A. Y.Hussein, and A. T.Sadiq, "Meerkat Clan-Based Feature Selection in Random Forest Algorithm for IoT Intrusion Detection," Iraqi Journal of Computer, Communication, Control and System Engineering, pp. 15–24, Sep. 2022, doi: 10.33103/uot.ijccce.22.3.2.*

[11] *M. F. Elrawy, A. I. Awad, and H. F. A. Hamed, "Intrusion detection systems for IoT-based smart environments: a survey," Journal of Cloud Computing, vol. 7, no. 1. Springer Verlag, Dec. 01, 2018. doi: 10.1186/s13677-018-0123-6.*

[12] *E. Gyamfi and A. Jurcut, "Intrusion Detection in Internet of Things Systems: A Review on Design Approaches Leveraging Multi-Access Edge Computing, Machine Learning, and Datasets," Sensors, vol. 22, no. 10. MDPI, May 01, 2022. doi: 10.3390/s22103744.*

[13] *A. Y. Hussein, and A. T. Sadiq, "Meerkat Clan-Based Feature Selection in Random Forest Algorithm for IoT Intrusion Detection," Iraqi Journal of Computers, vol. 22, no. 3, 2022, doi: 10.33103/uot.ijccce.22.3.2.*

[14] *S. Ardabili, A. Mosavi, and A. R. Varkonyi-Koczy, "Advances in Machine Learning Modeling Reviewing Hybrid and Ensemble Methods," 2019, doi: 10.20944/preprints201908.0203.v1.*

[15] *I. Ibrahim and A. Abdulazeez, "The Role of Machine Learning Algorithms for Diagnosing Diseases," Journal of Applied Science and Technology Trends, vol. 2, no. 01, pp. 10–19, Mar. 2021, doi: 10.38094/jastt20179.*

[16] *Y. Dhebar and K. Deb, "Interpretable Rule Discovery Through Bilevel Optimization of Split-Rules of Nonlinear Decision Trees for Classification Problems", vol. 51, no. 11, pp. 5573-5584 , 2022, doi: 10.1109/TCYB.*

[17] *W. Lin, Z. Wu, L. Lin, A. Wen, and J. Li, "An ensemble random forest algorithm for insurance big data analysis," IEEE Access, vol. 5, pp. 16568–16575, Aug. 2017, doi: 10.1109/ACCESS.2017.2738069.*

[18] R. M. Balabin, R. Z. Safieva, and E. I. Lomakina, "Comparison of linear and nonlinear calibration models based on near infrared (NIR) spectroscopy data for gasoline properties prediction," Chemometrics and Intelligent Laboratory Systems, vol. 88, no. 2, pp. 183–188, Sep. 2007, doi: 10.1016/j.chemolab.2007.04.006.

[19] K. Vembandasamyp, R. R. Sasipriyap, and E. Deepap, "Heart Diseases Detection Using Naive Bayes Algorithm," International Journal of Innovative Science, Engineering & Technology,vol. 2, no. 9, pp. 441-444 ,2015. [Online]. Available: www.ijiset.com

[20] R. Amjad and M. S. Croock, "Dominated destinations of tourist inside iraq using personal information and frequency of travel," Telkomnika (Telecommunication Computing Electronics and Control), vol. 17, no. 4, pp. 1723–1730, Aug. 2019, doi: 10.12928/TELKOMNIKA.V17I4.11956.

[21] S. Sperandei, "Understanding logistic regression analysis," Biochem Med (Zagreb), vol. 24, no. 1, pp. 12–18, 2014, doi: 10.11613/BM.2014.003.

[22] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," Front Neurorobot, vol. 7, no. DEC, 2013, doi: 10.3389/fnbot.2013.00021.

[23] T. Bräunl, B. McCane, M. Rivera, and X. Yu, Eds., Image and Video Technology, vol. 9431. in Lecture Notes in Computer Science, vol. 9431. Cham: Springer International Publishing, 2016. doi: 10.1007/978-3-319-29451-3.

[24] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in 2015 Military Communications and Information Systems Conference, MilCIS 2015 - Proceedings, Institute of Electrical and Electronics Engineers Inc., Dec. 2015. doi: 10.1109/MilCIS.2015.7348942.

[25] A. Y. Hussein, P. Falcarin, and A. T. Sadiq, "Enhancement performance of random forest algorithm via one hot encoding for IoT IDS," Original Research, vol. 9, no. 3, pp. 579–591, 2021.

[26] I. Ullah and Q. H. Mahmoud, "A Scheme for Generating a Dataset for Anomalous Activity Detection in IoT Networks."

[27] A. Alhowaide, I. Alsmadi, and J. Tang, "Towards the design of real-time autonomous IoT NIDS," Cluster Comput, pp. 1-14,2021, doi: 10.1007/s10586-021-03231-5.

[28] B. Roy, "School of Computing, Engineering and Mathematics A Deep Learning Approach for Intrusion Detection in Internet of Things using Bi-Directional Long Short-Term Memory Recurrent Neural Network,"pp. 1–6, 2018.

[29] A. Kim, M. Park, and D. H. Lee, "AI-IDS: Application of Deep Learning to Real-Time Web Intrusion Detection," IEEE Access, vol. 8, pp. 70245–70261, 2020, doi: 10.1109/ACCESS.2020.2986882.

[30] D. H. Abd, A. T. Sadiq, and A. R. Abbas, "Political arabic articles classification based on machine learning and hybrid vector," in CITISIA 2020 - IEEE Conference on Innovative Technologies in Intelligent Systems and Industrial Applications, Proceedings, Institute of Electrical and Electronics Engineers Inc., Nov. 2020. doi: 10.1109/CITISIA50690.2020.9371791.

[31] M. Ge, X. Fu, N. Syed, Z. Baig, G. Teo, and A. Robles-Kelly, "Deep learning-based intrusion detection for IoT networks," in Proceedings of IEEE Pacific Rim International Symposium on Dependable Computing, PRDC, IEEE Computer Society, Dec. 2019, pp. 256–265. doi: 10.1109/PRDC47002.2019.00056.