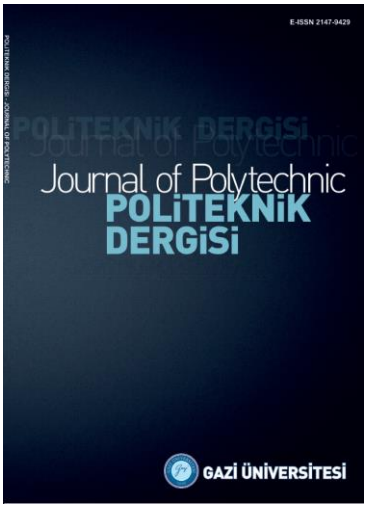


POLİTEKNİK DERGİSİ

JOURNAL of POLYTECHNIC

ISSN: 1302-0900 (PRINT), ISSN: 2147-9429 (ONLINE)
URL: http://dergipark.org.tr/politeknik

Detecting the cyber attacks on IoT-based network devices using machine learning algorithms

Makine öğrenimi algoritmaları kullanılarak IoT tabanlı ağ cihazlarına yönelik siber saldırıların tespiti

***Yazar(lar) (Author(s))**: M. Hanefi CALP[1*], Resul BUTUNER[2]*

ORCID[1]: 0000-0001-7991-438X

ORCID[2]: 0000-0002-9778-2349

To cite to this article: Calp M.H., Bütüner R., "Detecting the cyber attacks on ıot-based network devices using machine learning algorithms", ***Journal of Polytechnic***, 27(5): 1971-1989, (2024).

Bu makaleye şu şekilde atıfta bulunabilirsiniz: Calp M.H., Bütüner R., "Detecting the cyber attacks on ıot-based network devices using machine learning algorithms", ***Politeknik Dergisi***, 27(5): 1971-1989, (2024).

Erişim linki (To link to this article): http://dergipark.org.tr/politeknik/archive

DOI: 10.2339/politeknik.1340515

Detecting the Cyber Attacks on IoT-Based Network Devices Using Machine Learning Algorithms

Highlights

- *This study aims to detect cyber-attacks for security with ML algorithms by using data obtained from an IoT-based system.*
- *Artificial Neural Network (ANN), Random Forest (RF), K-Nearest Neighbor (KNN), Naive Bayes (NB), and Logistic Regression (LR) algorithms were used to create the models.*
- *The best performance to detect cyber-attacks was obtained using the RF algorithm with a rate of 99.6%.*

Graphical Abstract

In this study, a model that detects cyber-attacks to ensure security with machine learning (ML) algorithms was proposed by using the data obtained from the log records of an IoT-based system.

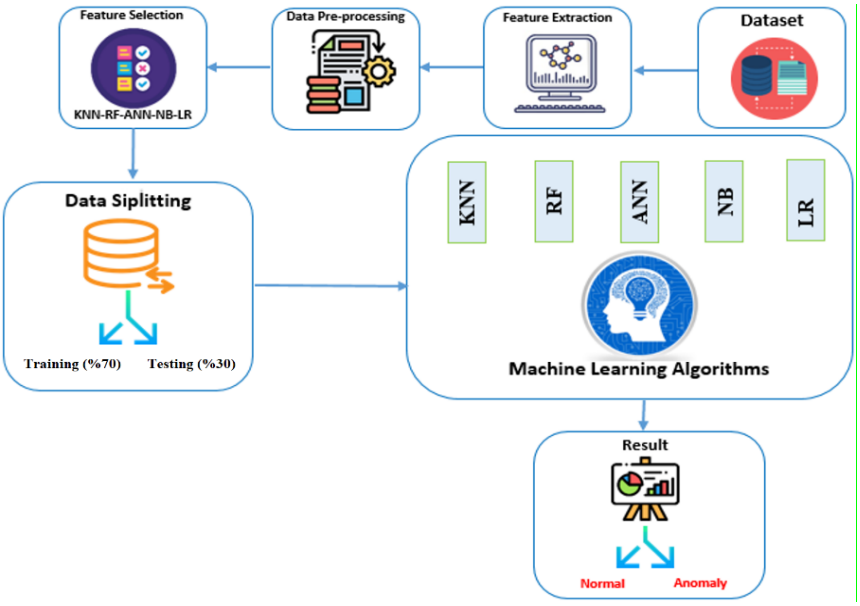


Figure. General design of the system

Aim

This study aims to detect cyber attacks on network devices with ML algorithms by using data obtained from an IoT-based system.

Design & Methodology

The data obtained from an IoT-based system and used in the creation of the models were pre-processed, all data were divided into training (70%) and testing (30%), and ML algorithms were used.

Originality

The models were created using 5 different machine-learning algorithms including KNN, RF, ANN, NB, and LR

Findings

The best performance to detect cyber-attacks was obtained using the RF algorithm with a rate of 99.6%. The performance of the NB algorithm is lower than that of other methods.

Conclusion

Conclusions showed that artificial intelligence algorithms were an effective method for attack detection and prevention in environments where IoT devices were present.

Declaration of Ethical Standards

The author(s) of this article declares that the materials and methods used in this study do not require ethical committee permission and/or legal-special permission.

Detecting the Cyber Attacks on IoT-Based Network Devices Using Machine Learning Algorithms

Araştırma Makalesi / Research Article

M. Hanefi CALP[1*], Resul BÜTÜNER[2]

[1]Faculty of Economics & Administrative Sciences, Department of Management Information Systems, Ankara Hacı Bayram Veli University, Türkiye

[2]Ankara Beypazarı Fatih Vocational and Technical Anatolian High School, Department of Computer, Ankara, Türkiye

(Geliş/Received : 10.08.2023 ; Kabul/Accepted : 06.10.2023 ; Erken Görünüm/Early View : 05.02.2024)

ABSTRACT

Today, the number and variety of cyber-attacks on all systems have increased with the widespread use of internet technology. Within these systems, Internet of Things (IoT)-based network devices are especially exposed to a lot of cyber-attacks and are vulnerable to these attacks. This adversely affects the operation of the devices in question, and the data is endangered due to security vulnerabilities. Therefore, in this study, a model that detects cyber-attacks to ensure security with machine learning (ML) algorithms was proposed by using the data obtained from the log records of an IoT-based system. For this, first, the dataset was created, and this dataset was preprocessed and prepared by the models. Then, Artificial Neural Network (ANN), Random Forest (RF), K-Nearest Neighbor (KNN), Naive Bayes (NB), and Logistic Regression (LR) algorithms were used to create the models. As a result, the best performance to detect cyber-attacks was obtained using the RF algorithm with a rate of 99.6%. Finally, the results obtained from all the models created were compared with other academic studies in the literature and it was seen that the proposed RF model produced very successful results compared to the others. Moreover, this study showed that RF was a promising method of attack detection.

Keywords: Internet of things, network devices, security, cyber-attack, machine learning.

Makine Öğrenimi Algoritmaları Kullanılarak IoT Tabanlı Ağ Cihazlarına Yönelik Siber Saldırıların Tespiti

ÖZ

Günümüzde internet teknolojisinin yaygınlaşmasıyla birlikte tüm sistemlere yönelik siber saldırıların sayısı ve çeşidi artmıştır. Bu sistemler içerisinde özellikle Nesnelerin İnterneti (IoT) tabanlı ağ cihazları çok sayıda siber saldırıya maruz kalmakta ve bu saldırılara karşı savunmasız kalmaktadır. Bu durum söz konusu cihazların çalışmasını olumsuz etkilemekte ve güvenlik açıkları nedeniyle veriler tehlikeye girmektedir. Bu nedenle bu çalışmada IoT tabanlı bir sistemin log kayıtlarından elde edilen veriler kullanılarak makine öğrenmesi (ML) algoritmaları ile güvenliği sağlamak için siber saldırıları tespit eden bir model önerilmiştir. Bunun için öncelikle veriseti oluşturulmuş ve bu veriseti ön işleme tabi tutularak modellere uygun olarak hazırlanmıştır. Ardından modelleri oluşturmak için Yapay Sinir Ağı (YSA), Rastgele Orman (RF), K-En Yakın Komşu (KNN), Naive Bayes (NB) ve Lojistik Regresyon (LR) algoritmaları kullanılmıştır. Sonuç olarak, siber saldırıları tespit etmede en iyi performans %99.6 ile RF algoritması kullanılarak elde edilmiştir. Son olarak oluşturulan tüm modellerden elde edilen sonuçlar literatürdeki diğer akademik çalışmalarla karşılaştırılmış ve önerilen RF modelinin diğerlerine göre oldukça başarılı sonuçlar ürettiği görülmüştür. Ayrıca, bu çalışma RF'nin gelecek vaat eden bir saldırı tespit yöntemi olduğunu göstermiştir.

Anahtar Kelimeler: Nesnelerin interneti, ağ cihazları, güvenlik, siber saldırı, makine öğrenimi.

1. INTRODUCTION

Computer systems, which have found their place in all areas of life, are widely used in different sectors and different ways. Most of these systems benefit from Internet technologies. With the use of Internet technology, the security level of the local network decreases and it becomes highly vulnerable to attacks. Thus, data is compromised in terms of privacy and usability. Unauthorized access and corruption of other hosts also lower the level of security on the network. Thus, there are many security vulnerabilities, and the type, size, and frequency of each cyber-attack are increasing [1-4].

Cyber-attacks not only try to obtain a system's login information but also perform much more dangerous processes such as unauthorized access, use, disclosure, destruction, modification, or damage. The priority in an attack is to access or seek information about a system on the network. Information about an attacker is possible by finding the list of open ports [5]. Cyber-attacks are constantly updating themselves with very complex algorithms. Cyber-attacks pose serious security hazards

*Sorumlu Yazar (Corresponding Author)
e-posta : hanefi.calp@hbv.edu.tr

and challenges. Therefore, they need a flexible, powerful, and reliable intrusion detection system [6]. In general, when cyber-attacks are detected on time, the damage to systems is tolerable and largely controllable [7].

In recent years, expenditures on cyber security technologies have been constantly increasing to provide a secure service to institutions and organizations. Security bugs can be prevented by using technologies such as user authentication and firewall data encryption. However, these technologies cannot perform a detailed analysis and therefore cannot reach the desired level of intrusion detection. More effective and sensitive systems have begun to be produced compared to traditional security methods with the increase in people, institutions, and applications using Internet technology. For this, Intrusion Detection and Prevention systems have been developed. The efficiency of these systems has been increased from ML methods, and it has started to be used widely, especially with IoT-based network technologies [7-12].

Intrusion Detection Systems (IDSs) aim to detect and prevent attacks outside or inside the network to be protected. IDSs contribute to both network and host-based security. IDSs are an essential part of system security. It not only helps to detect threats and attacks but also tries to maintain the safe state of the system. IDSs monitor the system or network for malicious activity. The primary role of IDSs is to detect attacks and respond accordingly. These systems are divided into two signature-based and anomaly-based. Signature-based systems are performed by storing previously seen and known attack types of a database. Anomaly-based systems, on the other hand, evaluate the anomalies of real-time packets of regular packets. That is, it tries to detect anomalies in the system. ML methods are generally used to detect these anomalies [4-13].

Network behavior data is collected and analyzed by some network equipment. The goal here is to detect potential threats or attacks lurking in network traffic. Traffic analysis in the network is done using anomaly detection and protocol analysis. This method is not effective in detecting unknown attacks. Heuristics are preferred to find unknown malicious activities. Stateful protocol analysis is the most powerful method. Because it has effects on the application layer, network layer, and transport layer. In recent years, deep learning approaches have been thought to improve intrusion detection techniques. However, there is not a sufficient number and variety of scientific studies to compare such approaches with open datasets. Common problems based on machine learning approaches are [6]:

- models produce highly inaccurate results due to a wider attack range,
- models cannot be generalized because only a single dataset is used to report the performance of the ML model,
- models studied so far failed to fully see today's massive network traffic, and
- finally, its inability to keep up speed and network size.

Therefore, all these mentioned situations and problems constitute the motivation of this study and reveal their importance. In this context, it was aimed to detect cyber-attacks for security with ML algorithms by using data obtained from an IoT-based system. In the second part of the study, the literature on the subject and all the details of the materials and methods used in the third part were given. In the fourth chapter, the results obtained from the models and the analysis and discussion of these results were given. Finally, in the fifth chapter, there were general conclusions drawn from the study and recommendations made within the scope of the study.

2. LITERATURE REVIEW

In the literature, many studies have been carried out on cyber security, IoT technology, and cyber-attack detection. In this context, this section was a summary of these studies. Ozgur and Erdem suggested that feature selection and classifier fusion weighting processes should be performed using a Genetic Algorithm (GA) in intrusion detection classification applications. A GA-based Feature Selection and Weighting system had been implemented on the dataset. Adaboost, Decision Tree (DT), LR, Pure Bayes, RF, Gradient Boosting, KNN, and ANN methods were used. The proposed method was compared with the results of other fusion methods (simple and probabilistic votes) and a single classifier. As a result, the proposed system was found to be more successful when compared with other previously published studies. The next study planned to apply the variable-length versions of the GA and other meta-heuristic methods for the same problem [14].

Demir proposed a powerful ML-based approach using the ISCX-2012 dataset. With this approach, cyber-attacks were detected with 100% accuracy. Feature and hyperparameter selection algorithms are used to improve the classification accuracy performance of the proposed method. The obtained results showed that the ML classifiers used for IDS provide superior performance. Better classification accuracies were obtained than in other studies using the same dataset. Moreover, these classification accuracies are achieved with fewer features (3 features), and the computational cost is reduced. While the best classification result was obtained with the KA classifier, the hyperparameter selection was made automatically with the Bayes algorithm [15]. Gazel and Bati aimed to find the best classification model in deep neural networks. The authors used Rmsprop, Sgd, Adam, Adagrad, and Nadam optimization methods, Tanh and ReLU activation functions, and neuron numbers. The best classification model was determined by comparing the performances of the model combinations created, and it was observed. In addition, it was stated that when working with the combination of different parameters of the optimization methods in the model created, a more suitable architecture of the dataset was obtained [16].

Pehlivanoglu et al. proposed the CSE-CIC-IDS2018 dataset and a single-two-level model for intrusion detection, and it was demonstrated that the classification performance could be increased. In the study, the dataset was handled by using Convolutional Neural Network (CNN), RF, Light Gradient Augmentation (LGBM), (CNN + RF), (LGBM + RF), and (RF+ RF) ML methods. With 98% accuracy and 0.86 macro F-score, the hybrid model (CNN + RF) was found to perform the best attack detection. In addition, hyperparameter optimization was performed with GridSearch, and the effect of the Synthetic Minority Oversampling Technique (SMOTE) and highly correlated features of detection were investigated. The Bi-Level method, in which CNN and RF methods are used together when the experimental results are analyzed, has the highest performance with a 0.86 F-score macro average. They suggested that different ML and deep learning methods should be tested for the hybrid model and a model that detects attacks simultaneously should be developed to increase its performance [17]. Cakir and Angin, the performance of Temporal Convolutional Networks (TCN), which is a deep learning method that has achieved high success, especially in the field of computer vision, in attack detection has been examined. The performance of TCN in both binary classification and anomaly detection problems is compared with repetitive neural networks and fully connected neural network methods, which have achieved high performance in many intrusion detection problems. The results show that TCN is at least as successful as LSTM in binary classification, which categorizes network traffic as normal and attacks. It has also been observed that it achieves high performance from many classical ML models. Finally, the authors planned to evaluate the effectiveness of TCN on different cyber-attack datasets and to reveal in which situations its use provides high performance [18].

Hatipoglu and Tunacan aimed to investigate the situation detection of cybercrime in Turkey and the solution methods produced in response to the types of cyber-attacks. As a methodology in the study, the literature review method was chosen. DoS and DDoS attacks from cyber-attack types and the Random Forest decision tree method were investigated. In the researched studies, the attack methods that are subject to the research and the methods used to detect and prevent them have been analyzed over the years. When looking at the types of cyber-attacks, it has been observed that DoS and DDoS attack types and the Random Forest decision tree method are the most examined and studied. In addition, especially in 2020, it has been observed that researchers prefer the systems they have developed independently of deep learning and ML techniques [19]. Aytan and Barisci aimed to detect DoS and Information Scanning attacks with ML algorithms. For this purpose, the Weka package program was used and the "KDD Cup'99" dataset, which is one of the commonly used datasets in applications related to intrusion detection systems, was used. As a result, the Back Propagation Algorithm has been used for large datasets and it has been determined that it is not suitable due to the long training period. The best detection was obtained with the Random Forest Algorithm with a success rate of 99%. The authors suggested that for large datasets, the Random Forest Algorithm can be used because the training time is very good and it gives a good detection, and learning percentage. They stated that the Backpropagation Algorithm can also be used for small datasets rather than large datasets [20].

Gurmen aimed to improve the methods used for the detection of attacks and to develop different methods. He examined these attacks under the headings of Denial of Service (DoS), LAN Login by Hijacking the Administrator Account (R2L), Scanning Information (Probe), and Upgrading the User Account to the Administrator Account (U2R). Classification models have been developed by using normalization, filtering feature selection, and proposed hybrid feature selection techniques for better and more effective intrusion detection solutions. The NSL-KDD dataset, which is frequently used in training IDSs, was used as a dataset. It has been tried to develop more effective intrusion detection methods by using the classification Naive Bayes, KNN, ANN, SVM, and DT which are methods of ML, and AdaBoosting one of the Ensemble Learning classification algorithms. As a result, the IDS overall success rate was evaluated and the classifier model with the highest average success rate was the AdaBoosting ensemble classifier with a rate of 99.7818% [21].

Karimipour et al. proposed a tool for anomaly detection. Here, the aim is to design a scalable anomaly detection engine that can distinguish a true failure from an intelligent cyberattack. At the same time, in the proposed method, feature extraction with Symbolic Dynamic Filtering (SDF) is applied to reduce the computational load. The results showed that the system achieved 99% accuracy, 98% TPR, and less than 2% FPR [22]. Kavousi-Fard et al. proposed a secure method to detect data integrity attacks against wireless sensor networks in microgrids and to at least minimize these attacks. An intelligent anomaly detection method is proposed to detect malicious attacks with different severity levels. The results obtained from these criteria and the confusion matrix support the accuracy and performance value of the proposed model [23].

Mousavinejad et al. developed a new cyber-attack detection method consisting of a prediction step and a measurement updated step in a networked control system. The forecast ellipsoid set was updated with sensor measurement data, a new forecast ellipsoid set was calculated, and intrusion detection on communication networks was provided. For this, recursive algorithms have been proposed. As a result, the authors consider it important to propose a new model that can detect cyber-attacks against NCSs [24]. AlZubi et al. proposed an Intrusion Detection Framework powered by cognitive ML for the secure sharing of health data, patient data security, and privacy in health networks. This

proposed approach is patient-centered, protecting data on trusted devices (such as end-user mobile phones and control data share access). This study enables us to analyze security threats and threat models for various cyber-physical system levels. In addition, the study also includes the difficulties encountered while taking the cyber-physical system development process. Experimental results show that our proposed model achieves a 96.5% attack prediction rate, 98.2% accuracy, 97.8% efficiency rate, 21.3% less delay, and 18.9% communication cost compared to other existing models [25]. Smys presented a method for DDOS attacks in a telecommunications network using a combination of neural networks and support vector machines. This method is the detection and classification of the attack. This study showed that the proposed method has 40% better accuracy than current methods [26].

Asharf generally examined the models developed for the detection of intrusions. It also extensively examined attacks on IoT systems originating from compromised IoT devices. This study was research on Machine Learning and DL-based Intrusion Detection techniques for IoT-based systems. IoT architecture, IoT system vulnerabilities, protocols, and attacks at the IoT protocol level were discussed in detail. This study sought to provide researchers with comprehensive and useful information on various security challenges and possible solutions faced by IoT systems and networks, focusing on intrusion detection based on ML and DL-based methods [27]. Rashid et al. investigated an attack and anomaly detection technique to defend against IoT cybersecurity threats in smart cities. For this, they used ML algorithms (ANN, SVM, RF, LR, DT, and KNN). In contrast to existing studies focusing on single classifiers, they have also explored ensemble methods such as boosting, bagging, and stacking to improve the performance of the detection system. The results showed that stacking classifiers can better detect cyber-attacks on the systems in smart cities [28].

Alsamiri and Alsubhi aimed to detect IoT network attacks quickly and effectively using various ML algorithms. They applied various ML algorithms independently of each other. Finally, seven widely used ML algorithms with different characteristics were applied to the data. The performance ratios were obtained according to these algorithms and the F-measure was given. According to experimental results, the F-measure had a value between 0 and 1; Naive Bayes 0.77; QDA 0.86; Random Forest 0.97; ID3, 0.97; AdaBoost 0.97; MLP was 0.83 and K Nearest Neighbors 0.99 [29]. Dutta et al. presented an ensemble method following the heap generalization principle using deep models such as a Meta Classifier (Logistical Regression), Long Short Term Memory (LSTM), and Deep Neural Network (DNN). The results of the proposed method in terms of statistical significance have been tested. In addition, the results were compared with state-of-the-art approaches to network anomaly detection [30].

Awan et al. predicted the application-layer DDoS attacks in real time with different ML models. Two ML approaches, Multi-Layer Perceptron (MLP) and RF were applied for the detection of Denial of Service (DoS) attacks. An average of 99.5% accuracy was achieved in models with and without big data approaches. However, the big data approach outperformed the non-big data approach due to the distributed in-memory Spark computations during training and testing time. An attack could be detected in a few milliseconds in real-time. It was reported that in the future Apache Spark will be evaluated along with other big data tools for the accuracy of ML models, training time, and test time [31]. Wu et al developed and integrated a physical data ML approach to detect Cyber-Physical attacks in the Cyber Manufacturing System (CMS). To test and demonstrate the physical data ML security approaches, they developed two examples of simulation and experimentation. The same three different ML algorithms applied to the random forest algorithm obtained the highest average accuracy of 91.1%. In addition, it was stated that it is not easy to detect cyber-physical attacks on the CMS environment, and more research is needed [32]. Savas and Savas classified URL addresses as harmful or not using machine learning algorithms. In that study, they utilized support vector machines, random forest, Gaussian Naive Bayes, logistic regression, k-nearest neighbors, decision trees, multilayer perceptrons, and XGBoost algorithms. Data was obtained via USOM, Alexa, and Phishtank to be used for training and testing purposes. As a result of the research, they reached a 99.8% accuracy rate [33].

3. MATERIAL and METHOD

In this section, first, detailed information was given about the acquisition and preparation of data to use it in the creation of models, and then the general design of the system, the algorithms used, and the criteria used to determine and evaluate the level of performance of the proposed models.

3.1. Obtaining and Preparing the Data

In research on Cyber Attack Detection, DARPA, KDD99, and NSL-KDD datasets were used [34, 35]. In this context, the first DARPA Cyber Attack Detection dataset was created in 1998 at the MIT Lincoln laboratory [36]. To make ML algorithms work better on the created KDD99 dataset, the repetitive records were deleted, the data size was reduced and the NSL-KDD dataset was created [37, 38].

The sizes and general characteristics of these datasets were given in Table 1.

Table 1. Datasets [39]

Name	Train Dimension	Test Dimension	Note
DARPA99	6.2 GB	3.67 GB	The original dataset. TCP/IP files
KDD99	4898431 samples	311029 samples	Feature extracted and preprocessed
CSE-CICIDS2018, CICIDS2017	2560176	2560176 samples	Updated version
NSL-KDD	18234 samples	7558 samples	Updated, size reduced

In this study, the NSL-KDD dataset was obtained by using the "Kaggle" website, which was open to everyone, for Cyber Attack Detection in IoT-based systems. Data preprocessing, denoising, and extraction were performed on this dataset. The dataset has a large data category of 25192 records. This dataset has undergone training, validation, and testing to create the model. In Table 2, the data dimensions of the dataset as training and testing were given.

Table 2. Training and test data numbers and percentages

	Number of Data	Percentage (%)
Training	17634	70
Testing	7558	30
Total	25192	100

All detailed information (attributes, name, type, description, and values) of the dataset was given in Table 3. The dataset consists of 42 columns, 41 features, and 1 target column. 12 of the 42 features of the dataset were categories and 30 of them were integer type data. The target column consists of 2 labels. These are (1) Anomaly and (2) Normal labels. It indicates that there was an attack on the IoT-based system when the “Anomaly” label was obtained as the output, and no cyber-attack occurred when the “Normal” label was present. According to the data from 41 properties in the dataset, the algorithm model estimates a value in the abnormal or normal category. The resulting output appears as a combination of 41 properties.

Table 3. Properties of the dataset

Attribute Number	Features	Type	Definition	Value
1	duration	numeric	Connection length	-
2	protocol_type	categorical	Protocol type	icmp,tcp, udp
3	service	categorical	Service type	http, ftp, smtp, ssh, dns, etc
4	flag	categorical	flag	oth,rej,rsto,rstos0,s0,s1,s2,s3,sf,sh
5	src_bytes	numeric	Data from source to destination	-
6	dst_bytes	numeric	Number of data bytes	-
7	land	categorical	If the source and destination IP are the same, if not 1, then 0	0,1
8	wrong_fragment	numeric	Incorrect shredding	-
9	urgent	categorical	Number of emergency packages	0,1
10	hot	numeric	the ”hot" indicator	-
11	num_failed_logins	numeric	Number of incorrect entries	-
12	logged_in	categorical	If the login is successful, if not 1, if not 0	0,1
13	num_compromised	numeric	Number of violations of privacy	-
14	root_shell	categorical	If the ”Root Shell" was obtained, if not 1, then 0	0,1
15	su_attempted	numeric	If the command ”Su Root" is entered, if not 1, then 0	-
16	num_root	numeric	Number of ”Root" accesses	-
17	num_file_creations	numeric	Number of file creation operations	-
18	num_shells	categorical	Number of Shell prompts	0,1

Continuation of Table 3

19	num_access_files	numeric	Number of access operations to control files	-
20	num_outbound_cmds	categorical	number of outgoing commands in an FTP session	0
21	is_host_login	categorical	If the entry is in the ”hot" list, if not 1, then 0	0
22	is_guest_login	categorical	If the input is "guest", if not 1, then 0	0,1
23	count	numeric	The number of the same connections to the same server as the two previous connections	-
24	srv_count	numeric	The number of the same connections to the same service as the two previous connections	-
25	serror_rate	numeric	Percentage of ”SYN" error connections	-
26	srv_serror_rate	numeric	Percentage of connections to the same service	-
27	rerror_rate	numeric	Percentage of ”REJ" error links	-
28	srv_rerror_rate	numeric	The number of the same connections to the same service as the two previous connections	-
29	same_srv_rate	numeric	Percentage of connections to the same service	-
30	diff_srv_rate	numeric	Percentage of connections to different services	-
31	srv_diff_host_rate	numeric	Percentage of connections to different services	-
32	dst_host_count	numeric	sum of connections to the same destination IP address	-
33	dst_host_srv_count	numeric	sum of connections to the same destination port number	-
34	dst_host_same_srv_rate	numeric	the percentage of connections that were to the same service, among the connections aggregated in dst_host_count (32)	-
35	dst_host_diff_srv_rate	numeric	the percentage of connections that were to different services, among the connections aggregated in dst_host_count (32)	-
36	dst_host_same_src_port_rate	numeric	the percentage of connections that were to the same source port, among the connections aggregated in dst_host_srv_count (33)	-
37	dst_host_srv_diff_host_rate	numeric	the percentage of connections that were to different destination machines, among the connections aggregated in dst_host_srv_count (33)	-
38	dst_host_serror_rate	numeric	the percentage of connections that have activated the flag (4) s0, s1, s2 or s3, among the connections, aggregated in dst_host_count (32)	-
39	dst_host_srv_serror_rate	numeric	the percent of connections that have activated the flag (4) s0, s1, s2 or s3, among the connections aggregated in dst_host_srv_count (33)	-
40	dst_host_rerror_rate	numeric	the percentage of connections that have activated the flag (4) REJ, among the connections aggregated in dst_host_count (32)	-
41	dst_host_srv_rerror_rate	numeric	the percentage of connections that have activated the flag (4) REJ, among the connections aggregated in dst_host_srv_count (33)	-
42	class	categorical		anomaly, normal

In Figure 1, the total number of data and the total dataset according to the target tag were given. Accordingly, there are 13449 “Normal” and 11743 “Anomaly” data. In Figure 2, the correlation graph between the protocol types according to the target tags in the dataset was given. Here, according to the dataset, there are 1655 data in the "ICMP" protocol, 20526 in the "TCP" protocol, and 3011 in the "UDP" protocol. The data was balanced according to normal and abnormal results in the TCP protocol. According to the obtained data, the dataset was created. Accordingly, data were obtained according to normal and abnormal results in ICMP and UDP protocol. There is no imbalance in the data set.

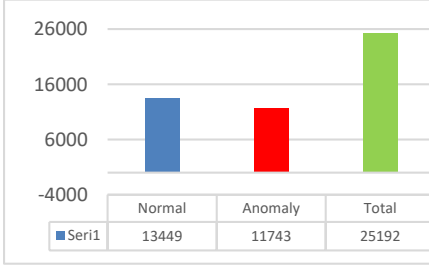


Figure 1. Data counts by target labels

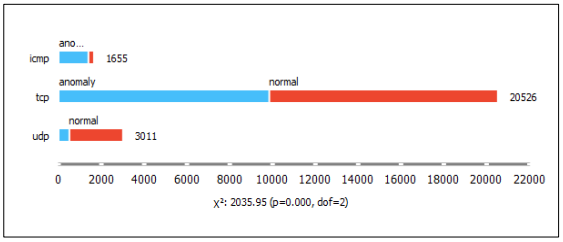


Figure 2. Correlation plot of target tags by protocol type

3.2. General Design of the System

This section contains all the details of the general structure of the study. In this context, according to the block design of the proposed system (Figure 3), firstly, the data used in the creation of the models were pre-processed. Then, all data were divided into training (70%) and testing (30%) and subjected to the training process. Finally, models were created using 5 different machine-learning algorithms: KNN, RF, ANN, NB, and LR. Two types of target label results were obtained "Normal" and "Anomaly" by applying the dataset to these models.

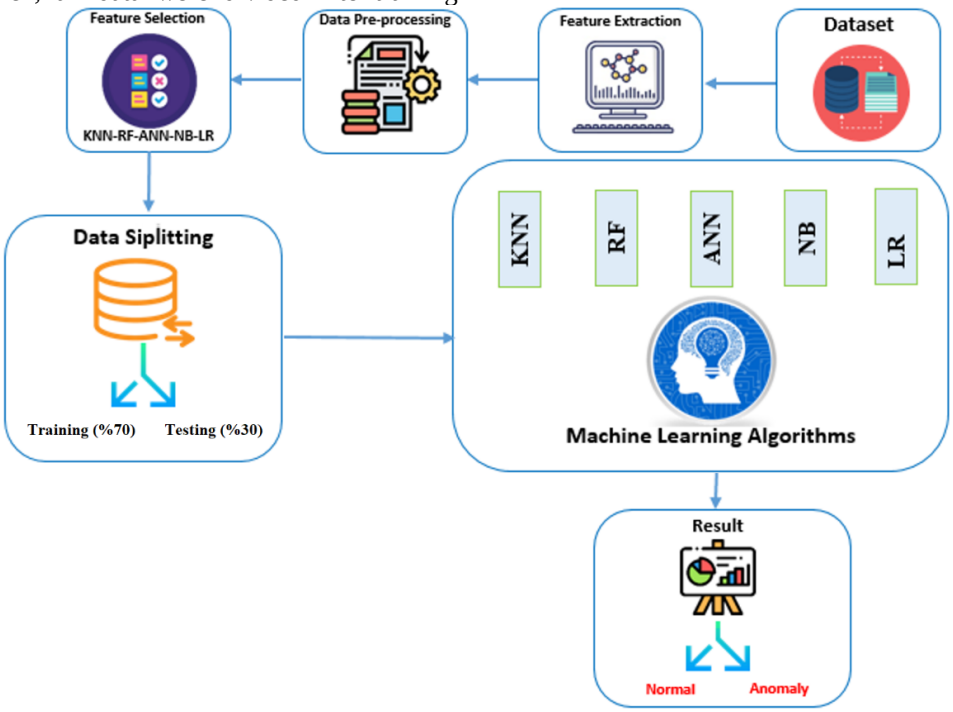


Figure 3. The general design of the system

Another model of the process from creating models to obtaining results for cyber-attack detection in IoT-based network devices was given in Figure 4. When Figure 4 was examined, it was seen that the data was passed through the training and testing phase, the models were created according to the determined features, and the results were estimated using these models.

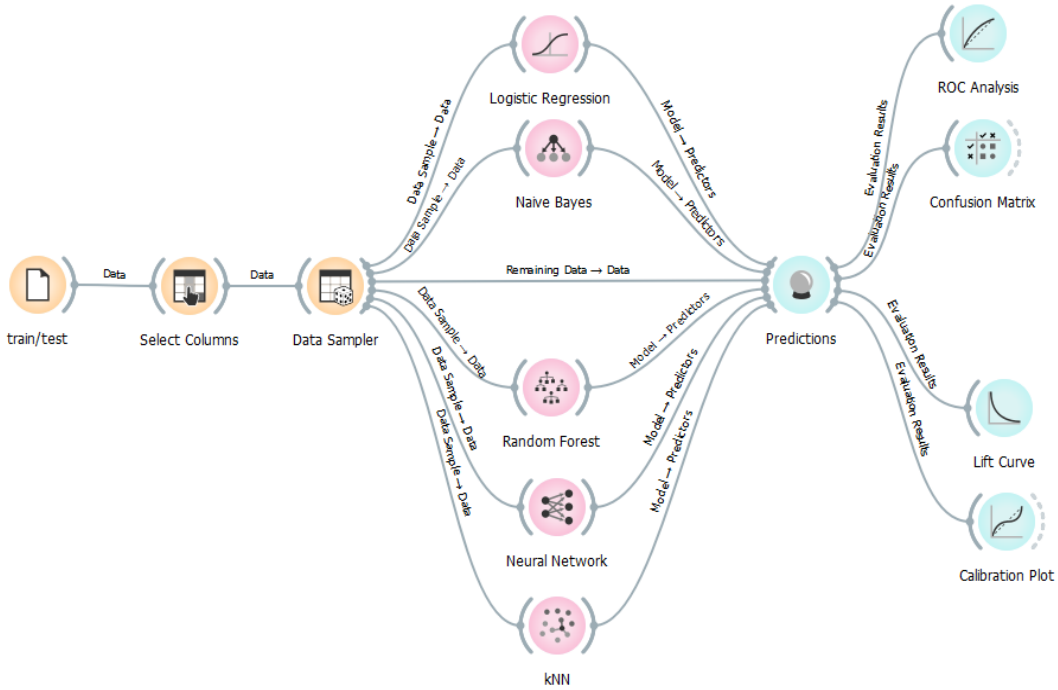


Figure 4. The process of creating the models and obtaining the results

3.3. Algorithms Used

KNN, RF, ANN, NB, and LR algorithms were used to create the Cyber Attack Detection model in the IoT-based system. In this section, brief information about these algorithms was given.

3.3.1. KNN algorithm

The KNN algorithm is one of the supervised learning algorithms that are easy to implement. KNN finds the nearest neighbors according to distances between neighbors and labels the data according to classes. The main issue in classification is to look at the properties of the objects and determine which class the objects belong to [40-42]. The KNN algorithm compares the data in the training set with each new data in the group and performs the classification process. Each sample in the training set represents a point in space. When a new sample joins the space, the class of the new sample is determined by determining the k samples in the training set closest to the new sample [43].

3.3.2. RF algorithm

RF is a classification method that includes the voting method. It is formed by collecting more than one DT. RF is an ensemble method that makes predictions based on the results of a collection of DT. Resampling is used to create each tree in the "forest" using the bootstrap approach. A random subset of features is selected at each node split, and the selection of the split variable takes place on this subset. RF is used as an improved version of the bagging method by adding the randomness feature. [44-47].

3.3.3. ANN algorithm

ANNs are a system that is widely used today, can perform learning functions by making use of experiments, and can predict. ANN, which is a different computation technique from the traditional computation technique, has a structure that keeps up with its environment and is adaptable. ANNs are used effectively in many different fields that make decisions in uncertain situations such as robotics, prediction, pattern recognition, fingerprint recognition, job scheduling and quality control, system modeling, finance applications, image processing, industrial applications, and defense applications [48-50]. In the ANN method, all data is transmitted to the network starting from the input layer. Then, this data is processed in the middle layers and finally transmitted to the output layer [51].

3.3.4. NB algorithm

NB Classification is an adaptation of Bayesian theory to ML algorithms with a high success rate, which was introduced for calculating conditional probabilities. In other words, it is the process of calculating probabilities according to NB Bayesian theory and classifying the desired variable according to the highest probability [52]. NB technique is based on an approach that can produce different results according to statistical values. It can be said that this technique consists of the integration of the decision tree model and the Bayes rule. The algorithm accepts that the features in the data belong to a certain class and performs the classification process by considering the most accurate or most appropriate label [53, 54].

3.3.5. LR algorithm

Logistic regression is an alternative to linear regression analysis because the normality assumption is broken. Logistic regression is aimed at performing mathematical modeling to describe the relationship between independent variables and two or multi-class categorical dependent variables [55, 56].

3.4. Performance Evaluation Criteria

Some criteria were used to evaluate the performance and success of the models created as a result of the classification process after applying ML algorithms to the dataset. These were Accuracy, Sensitivity, Specificity, Receiver Operating Characteristics (ROC), Recall, Precision, and F-Measure. These criteria aimed to compare the actual values of the model with the estimated values. The Confusion Matrix given in Table 4 was used to calculate the criteria. The Confusion Matrix was schematized as follows:

Table 4. Confusion Matrix

Real Class		**Estimated Values**	
		Anomaly	**Normal**
Estimated Class	**Anomaly**	True Positive (TP)	False Positive (FP)
	Normal	False Negative (FN)	True Negative (TN)

Among the actual and estimated data in this dataset [57,58];

TP: True-positive value is the positive prediction of the actual positive data.

FN: False-negative value is a false-positive prediction of data that is actual negative.

FP: False-positive value is the correct guess of the actual negative data.

TN: True-negative value is the correct estimation of the actual negative data.

The success of the classifier can be found in different ways. For example, in the Cyber Attack Detection classification, the output consists of two classes (Normal, Anomaly). While the classifier correctly classifies many samples, it can mark some attacks as Normal and some normal cases as “Anomaly”. The success of the model was evaluated according to the results of the confusion matrix by using ML algorithms. The success rates of each algorithm were calculated. The formulas for the content and calculation of these criteria were given in Equations 1-6.

Accuracy: It expresses the ratio of all correctly predicted values to all results.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Sensitivity: Sensitivity and TP rate (True positive rate) express the ratio of correctly predicted positive values to all positive values. In other words, it shows how many of those who are actually sick are detected. It indicates the probability of a positive decision being correct.

$$Sensitivity = TP\ Rate = \frac{TP}{TP + FP} \quad (2)$$

Specificity: It expresses the ratio of correctly predicted negative values to all negative values. It shows how many of the actually healthy ones can be detected correctly (invisible-normal). It shows the probability that a negative decision is correct.

$$Specificity = \frac{TN}{TN + FP} \quad (3)$$

ROC: ROC curve is a probability curve and the area under the curve is defined as Auc (area under the curve). It shows how well you can separate classes for AUC classification. It takes a value between 0-1 and it can be said that the classification performance increases as the value approaches 1, and the performance of the model is poor as the AUC value decreases and it makes random predictions. At the same time, the ROC curve shows the relationship between sensitivity and specificity values in the test.

Recall: It expresses the ratio of correctly predicted positive values to all true positive class values.

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

Precision: It is the ratio of the correctly predicted positive class value to all positively predicted class values.

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

F-Measure: It is the criterion used to evaluate the sensitivity and precision criteria together. The F-Measure is found by calculating the harmonic mean of these two criteria.

$$F - Measure = 2 * \frac{TP * Precision}{TP + Precision} \quad (6)$$

4. RESULTS AND DISCUSSION

In this section, the models created for five different methods and the classification results of these models were given. In addition, the results obtained from the models were compared, analyzed, and discussed.

4.1. KNN Model

The KNN model was created using the nearest neighbor value of "5", its metric "Euclidean", its weight "Distance", 17634 data for training, and 7558 data for testing (Table 5). According to these parameters and data, the test success of the model was 98.7%.

Table 5. KNN algorithm parameters

Nearest Neighbor	Metric	Weight	Train	Test	Test Accuracy
5	Euclidean	Distance	17634	7558	98.7%

During the creation of the KNN model, the distances were calculated by taking the dataset. Then, the model was applied by finding the nearest neighbors and was classified as to whether there was a cyber-attack on the data. The model classifies with the label "Anomaly" if there was an attack, and "Normal" if there was no attack. The KNN model according to the output layers was given in Figure 5. According to the KNN model, the dataset with the class label "Anomaly" was estimated with a rate of 98.5% and the dataset with "Normal" with a rate of 1.5%. The dataset with the class label "Normal" was estimated with a rate of 98.8% and the dataset with "Anomaly" with a rate of 1.2%.

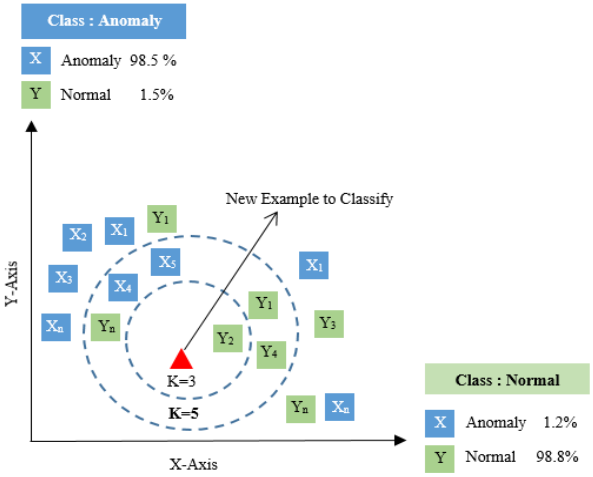


Figure 5. KNN model

The Confusion Matrix showing the current situation in the dataset and the number of correct and incorrect predictions of the KNN classification model was given in Table 6. The total correct numbers and ratios of each class were also shown. Considering the results of the evaluation criteria according to the KNN algorithm (Table 7), the precision was 0.987, the sensitivity (recall) was 0.987, the F1 Score was 0.987, and the classification success (CA) was 0.987, the accuracy value was 0.995. The ROC accuracy graph according to the KNN algorithm was given in Figure 6, and it was seen that the model achieved success above 0.98.

Table 6. KNN algorithm classification performance

		Predicted		
		Anomaly	**Normal**	∑
Actual	**Anomaly**	98.5%	1.2%	3523
	Normal	1.5%	98.8%	4034
	∑	3530	4027	7557

Table 7. Evaluation criteria according to the KNN algorithm

Evaluation Metrics					
Model	**Precision**	**Recall**	**F1 Score**	**CA**	**AUC**
KNN	0.987	0.987	0.987	0.987	0.995

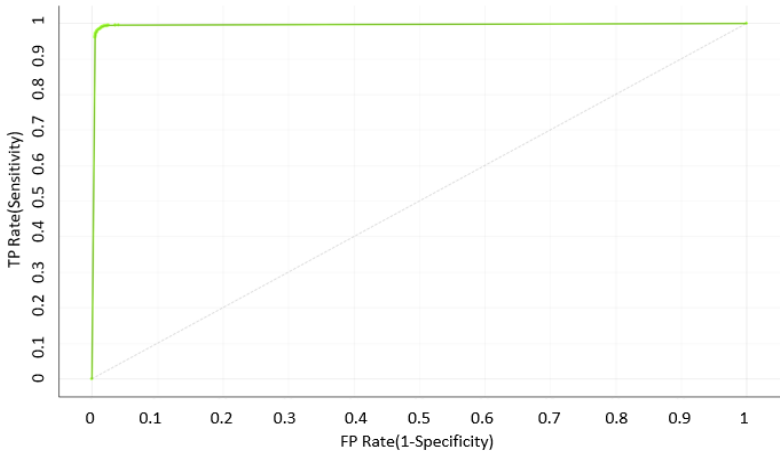


Figure 6. ROC curve of KNN algorithm according to classes

4.2. RF Model

The RF model was created using 20 trees, 5 predictions per division, 17364 data for training, and 7558 data for testing (Table 8). According to these parameters and data, the test success of the model was 99.6%.

Table 8. RF algorithm parameters

Trees	Predictors Per Split	Train	Test	Test Accuracy
20	5	17634	7558	99.6%

According to the model, the final result was produced by averaging the results. The RF algorithm model created in the study was given in Figure 7.

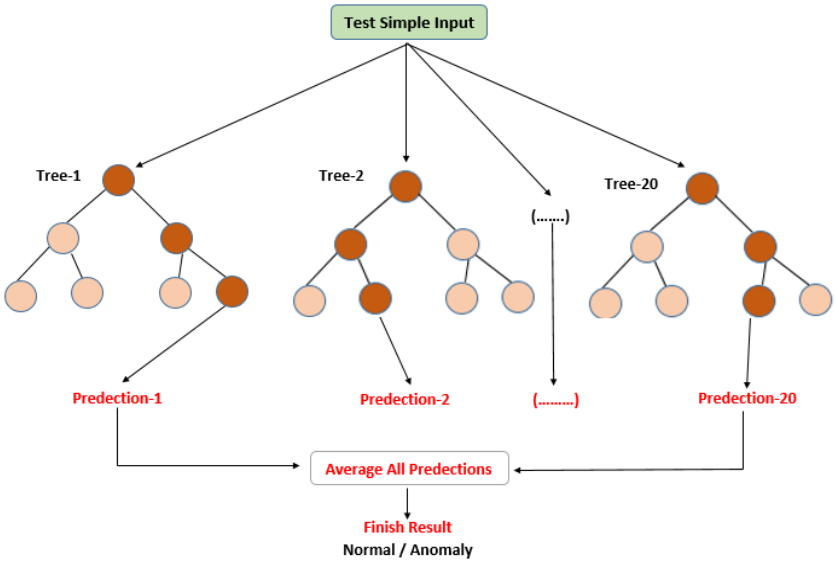


Figure 7. RF model

The Confusion Matrix showing the current situation in the dataset and the number of correct and incorrect predictions of the RF classification model was given in Table 9. The total correct numbers and ratios of each class were also shown. Looking at the results of the evaluation criteria according to the RF algorithm (Table 10), the precision was 0.996, the sensitivity (recall) was 0.996, the F1 Score was 0.996, and the classification success (CA) was 0.996, the accuracy value was 1.000. The ROC accuracy graph according to the RF algorithm was given in Figure 8, and it was seen that the model achieved success above 0.99.

Table 9. RF algorithm confusion matrix results

		Predicted		
		Anomaly	**Normal**	∑
Actual	**Anomaly**	99.9 %	0.6 %	3523
	Normal	0.1 %	99.4 %	4034
	∑	3501	4056	7557

Table 10. Evaluation criteria according to the RF algorithm

Evaluation Metrics					
Model	**Precision**	**Recall**	**F1 Score**	**CA**	**AUC**
RF	0.996	0.996	0.996	0.996	1.000

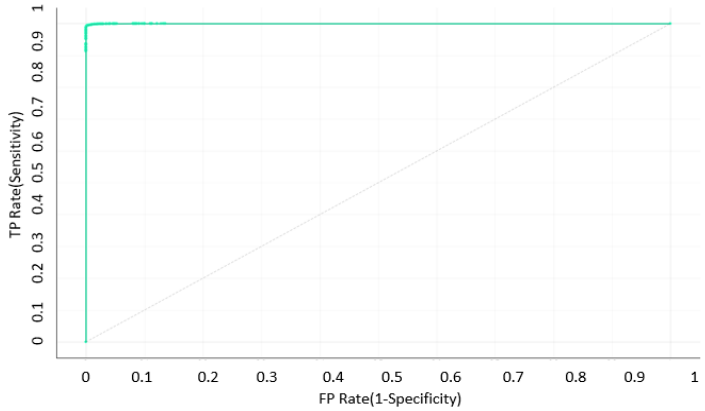


Figure 8. ROC curve according to RF algorithm classes

4.3. ANN Model

In the ANN model, there were 3 hidden layers and a total of 15 neurons, 3, 5, and 7 in each hidden layer (Table 11). The Activation function that created the model was "ReLU", the optimization method was "Adam", and the maximal number of iterations was 200. 17364 data were used for training and 7558 data were used for testing. According to these parameters and data, the test success rate was 98.3%.

Table 11. ANN algorithm parameters

Hidden Layers	**Neurons in Hidden Layers**	**Activation**	**Optimization Method**	**Maximal Number of Iteration**	**Train**	**Test**	**Test Accuracy**
3	3,5,7 =15	ReLU	Adam	200	17634	7558	98.3%

When calculated with Eq.1 in the three hidden layers ANN model, there were a total of 204 parameters to be learned; (3+5+7+2=17) 17 neurons and ([41x3]+[3x5]+[5x7]+[7x2]=123+15+35+14= 187) 187 weight values.

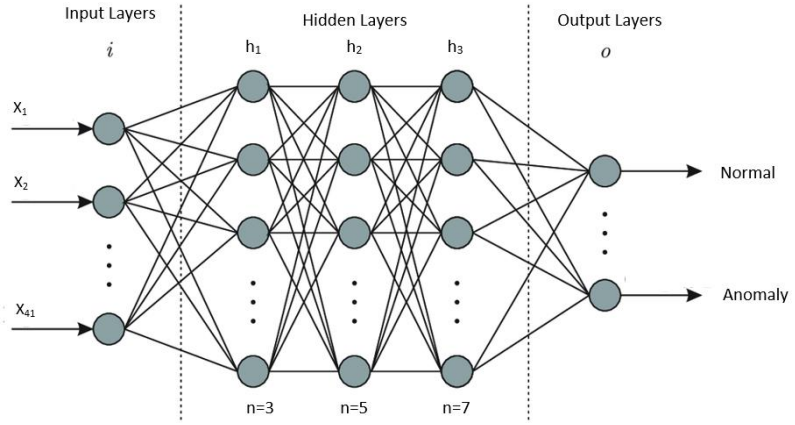


Figure 9. ANN model

The Confusion Matrix showing the current situation in the dataset and the number of correct and incorrect predictions of the ANN classification model was given in Table 12. The total correct numbers and ratios of each class were also shown. Considering the results of the evaluation criteria according to the ANN algorithm (Table 13), the precision was 0.983, the sensitivity (recall) was 0.983, the F1 score was 0.983, and the classification success (CA) was 0.983, and the accuracy value was 0.996. The ROC accuracy graph according to the ANN algorithm was given in Figure 10, and it was that the model achieved success above 0.98.

Table 12. Confusion matrix results of the ANN algorithm

		Predicted		
		Anomaly	**Normal**	∑
Actual	**Anomaly**	97.5 %	0.9 %	3523
	Normal	2.5 %	99.1 %	4034
	∑	3577	3980	7557

Table 13. Evaluation criteria according to the ANN algorithm

Evaluation Metrics					
Model	**Precision**	**Recall**	**F1 Score**	**CA**	**AUC**
ANN	0.983	0.983	0.983	0.983	0.996

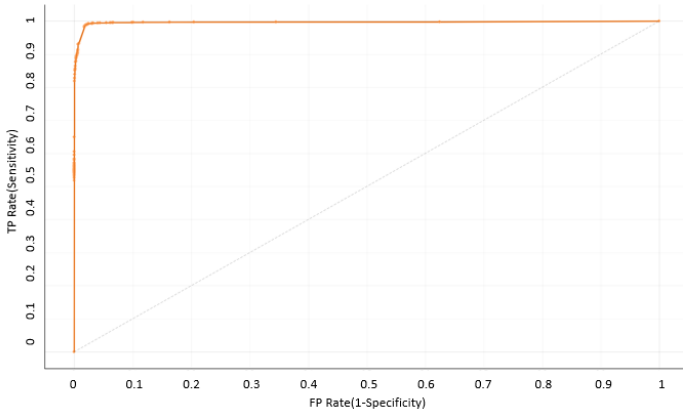


Figure 10. ROC curve of ANN algorithm according to classes

4.4. NB Model

The NB model was created using 17364 data for training and 7558 data for testing (Table 14). According to these parameters and data, the test success was 91.8%. In the statistical literature, Naive Bayesian models are known under various names, including simple Bayesian and independence Bayesian. The binary model was created under simple bayesian. It has been found that the Naive Bayesian algorithm is suitable for binary output type as output in simple bayesian model.

Table 14. NB algorithm parameters

Train	Test	Test Accuracy
17634	7558	91.8%

According to the NB algorithm model, success rates of 87.9% for the Normal class label and 97.6% for the Anomaly class label were obtained (Figure 11).

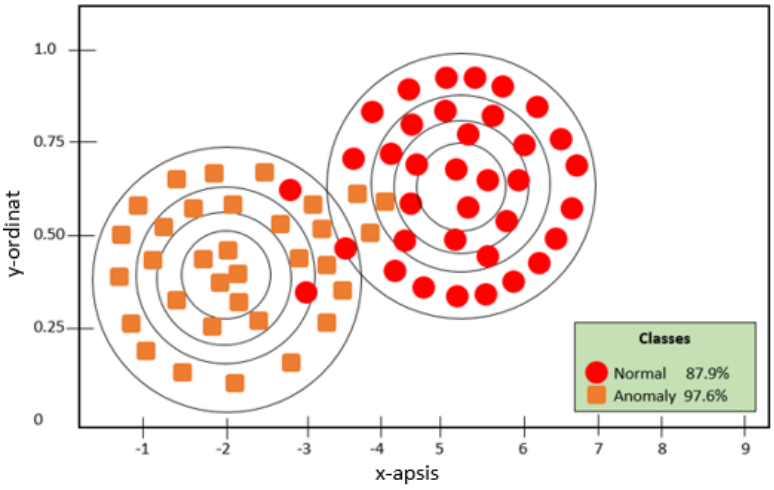


Figure 11. NB model

The Confusion Matrix showing the current situation in the dataset and the number of correct and incorrect predictions of the NB classification model was given in Table 15. The total correct numbers and ratios of each class were also shown. Considering the results of the evaluation criteria according to the NB algorithm (Table 16), precision was 0.924, sensitivity (recall) was 0.918, F1 Score was 0.917, classification success (CA) was 0.918, and accuracy value was 0.981. The ROC accuracy graph according to the NB algorithm was given in Figure 12, and it was seen that the model achieved success above 0.91.

Table 15. The confusion matrix of the NB algorithm

		Predicted		
		Anomaly	**Normal**	∑
Actual	**Anomaly**	97.6 %	12.1 %	3523
	Normal	2.4 %	87.9 %	4034
	∑	3048	4509	7557

Table 16. Evaluation criteria according to the NB algorithm

Evaluation Metrics					
Model	**Precision**	**Recall**	**F1 Score**	**CA**	**AUC**
NB	0.924	0.918	0.917	0.918	0.981

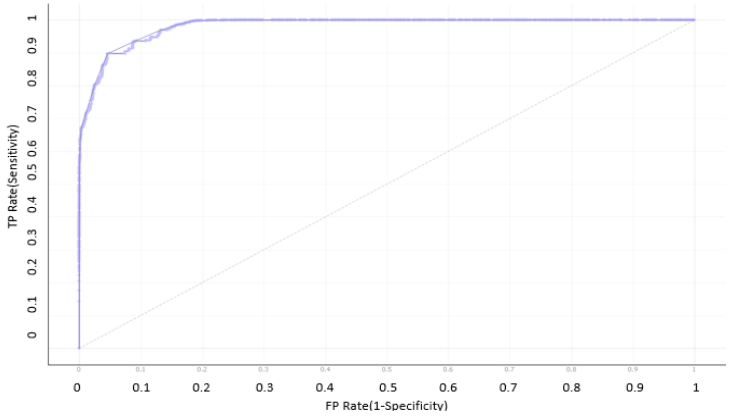


Figure 12. ROC curve of NB algorithm according to classes

4.5. LR Model

In the LR model, both variable selection and adjustment were made to increase the prediction accuracy and interpretability of the model produced by Lasso Regression. In Table 17, Lasso (Regulation Type) was selected according to the LR model and the test success was 97.2%.

Table 17. LR algorithm parameters

Regularization Type	**Train**	**Test**	**Test Accuracy**
Lasso (L1)	17634	7558	97.2 %

Figure 13 shows the graph according to the binary classification of the LR model used in the study. In the LR model, the threshold value was determined to determine the class to which the data belongs. It was divided into classes based on a threshold. There were two classes in the study, "Normal" and "Anomaly". The success results were achieved as 96.6% in the Normal class and 98% in the Anomaly class in the LR model by taking values smaller or larger than the threshold value. As seen in Figure 13, a binary LR algorithm was applied in the study. The activation function used was the sigmoid function.

According to the binary LR algorithm given in Figure 13, it was given 41 input layers and was classified into the output layers (Normal and Anomaly) in Figure 14. Also, each neuron in the network can be considered an LR; the input includes weights and bias, and a dot product was performed on all of them before applying any nonlinear functions. The last layer of a neural network was a basic linear model (maximum).

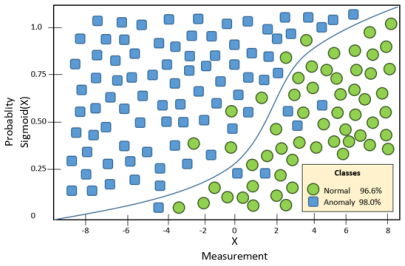


Figure 13. LR graph

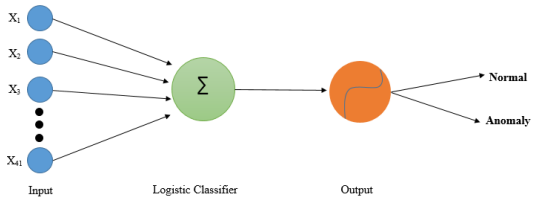


Figure 14. LR model

The Confusion Matrix showing the current situation in the dataset and the number of correct and incorrect predictions of the LR classification model was given in Table 18. The total correct numbers and ratios of each class were also shown. Considering the results of the evaluation criteria according to the LR algorithm (Table 19), precision was 0.972, sensitivity (recall) was 0.972, F1 Score was 0.972, classification success (CA) was 0.972, and accuracy value was 0.995. The ROC accuracy graph according to the LR algorithm was given in Figure 15, and it was seen that the model achieved success above 0.97.

But as the number of classifiers increases, these values (accuracy rate and F1-Value) decrease. ROC graph was created according to two classifications.

Table 18. The confusion matrix of the LR algorithm

		Predicted		
		Anomaly	**Normal**	∑
Actual	**Anomaly**	98 %	3.4 %	3523
	Normal	2 %	96.6 %	4034
	∑	3453	4104	7557

Table 19. Evaluation criteria according to the LR algorithm

Evaluation Metrics					
Model	**Precision**	**Recall**	**F1 Score**	**CA**	**AUC**
LR	0.972	0.972	0.972	0.972	0.995

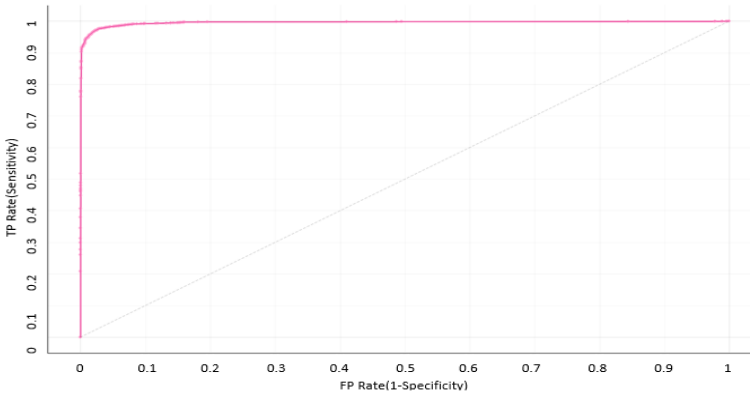


Figure 15. ROC curve of LR algorithm according to classes

4.6. Comparison and Analysis of Models

In this section, the findings obtained from the models created using KNN, RF, ANN, NB, and LR algorithms were compared and analyzed. According to the aforementioned findings (Figure 16), it was seen that the most successful results for cyber-attack detection were obtained by using the RF algorithm, one of the ML algorithms.

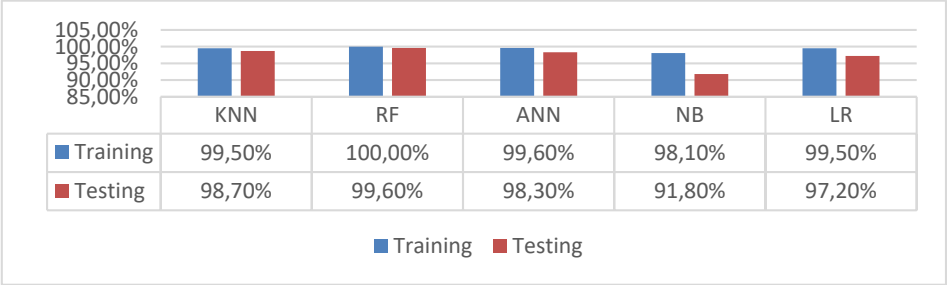


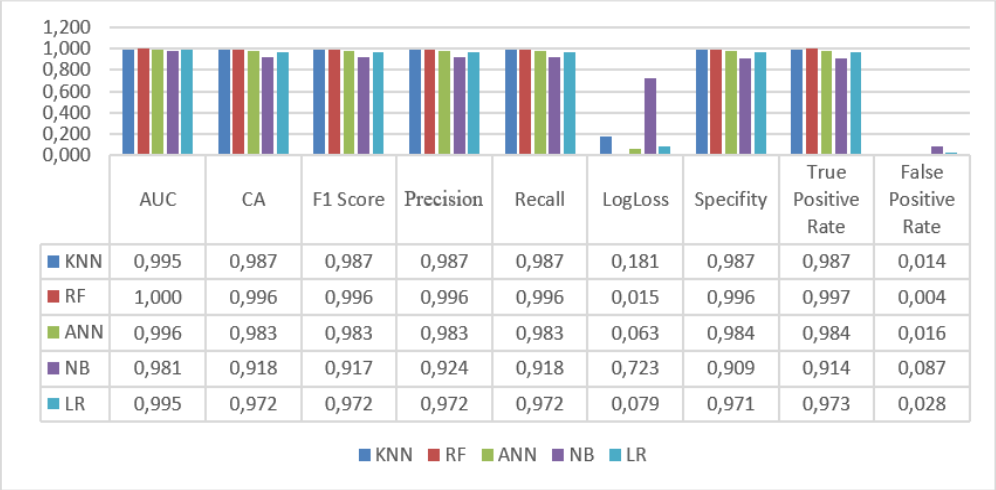
Figure 16. Comparison of algorithm results

In this process, KNN, ANN, NB, LR, and RF models were tested using real data. Experimental results obtained from randomly determined data were given in Table 20. Considering the experimental results, the best success rate and the lowest loss rate were obtained by using the RF algorithm.

Table 20. Experimental results from models

No	KNN	Loss (KNN)	ANN	Loss (ANN)	NB	Loss (NB)	LR	Loss(LR)	RF	Loss (RF)
1	99.00	1.00	98.40	1.60	90.40	9.60	96.40	3.60	99.80	0.20
2	99.41	0.59	99.97	0.03	90.47	9.53	97.47	2.53	99.47	0.53
3	99.14	0.86	98.32	1.68	87.32	12.68	98.32	1.68	99.92	0.08
4	96.60	3.40	98.75	1.25	90.75	9.25	98.75	1.25	99.75	0.25
5	99.15	0.85	97.86	2.14	89.56	10.44	97.56	2.44	99.56	0.44
6	98.96	1.04	98.30	1.70	98.30	1.70	96.30	3.70	98.80	1.20
7	98.60	1.40	97.41	2.59	96.41	3.59	97.41	2.59	99.71	0.29
8	99.60	0.40	98.46	1.54	90.56	9.44	97.56	2.44	99.56	0.44
9	97.40	2.60	98.20	1.80	90.20	9.80	96.20	3.80	99.90	0.10
10	98.60	1.40	97.43	2.57	88.43	11.57	97.43	2.57	99.63	0.37
Average	98.65	1.35	98.31	1.69	91.24	8.76	97.34	2.66	99.61	0.39

In Figure 17, evaluation metrics for ML models created for cyber-attacks in the IoT-based system were given. According to the RF algorithm, the classification accuracy (CA) was 98.7%, AUG was 99.5%, TPR was 98.7, and FPR was 0.14%, and the most successful results were obtained. NB algorithm obtained the classification accuracy (CA-91.8%, AUG (98.1%), True Positive Rate (TPR-91.4%), and False Positive Rate (FPR-0.87%), but more unsuccessful results were obtained compared to other algorithms.



	AUC	CA	F1 Score	Precision	Recall	LogLoss	Specifity	True Positive Rate	False Positive Rate
KNN	0,995	0,987	0,987	0,987	0,987	0,181	0,987	0,987	0,014
RF	1,000	0,996	0,996	0,996	0,996	0,015	0,996	0,997	0,004
ANN	0,996	0,983	0,983	0,983	0,983	0,063	0,984	0,984	0,016
NB	0,981	0,918	0,917	0,924	0,918	0,723	0,909	0,914	0,087
LR	0,995	0,972	0,972	0,972	0,972	0,079	0,971	0,973	0,028

Figure 17. Evaluation metrics by models

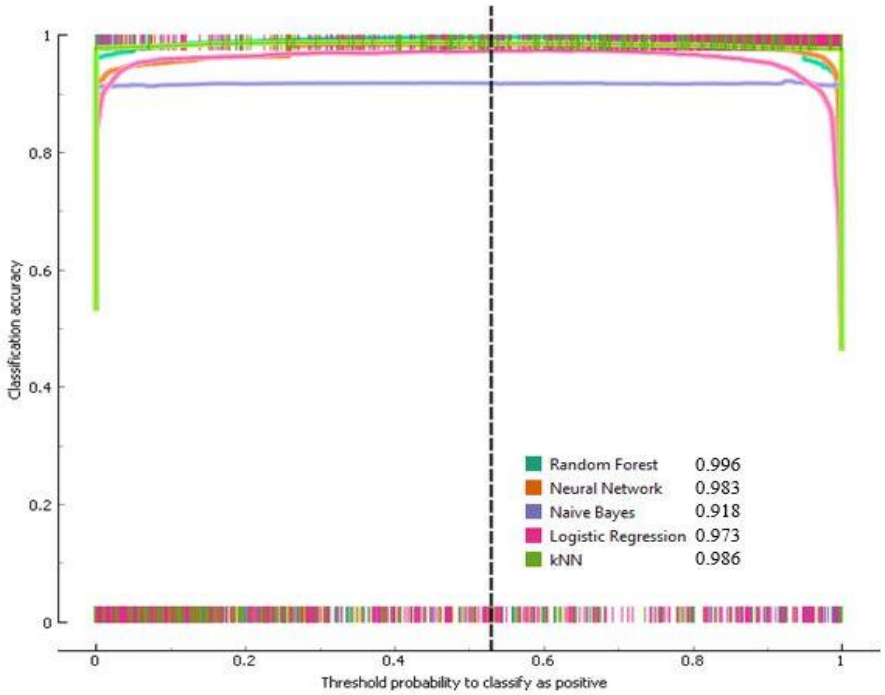


Figure 18. Classification accuracy by models

The calibration chart for all models was given in Figure 18. Looking at the values, it was seen that the RF algorithm gives the most successful results.

The ROC curve for the performance indicators of all the algorithms used in the study was given in Figure 19. The accuracy rates of the algorithms used in the study were plotted according to TP and FP. Here, TP was given as sensitivity, and FP was given as specificity (1-specificity). The target class for generating the ROC graph was normal and abnormal. Costs were the FP = 500, FN = 500 and Target probability: 53.0%.

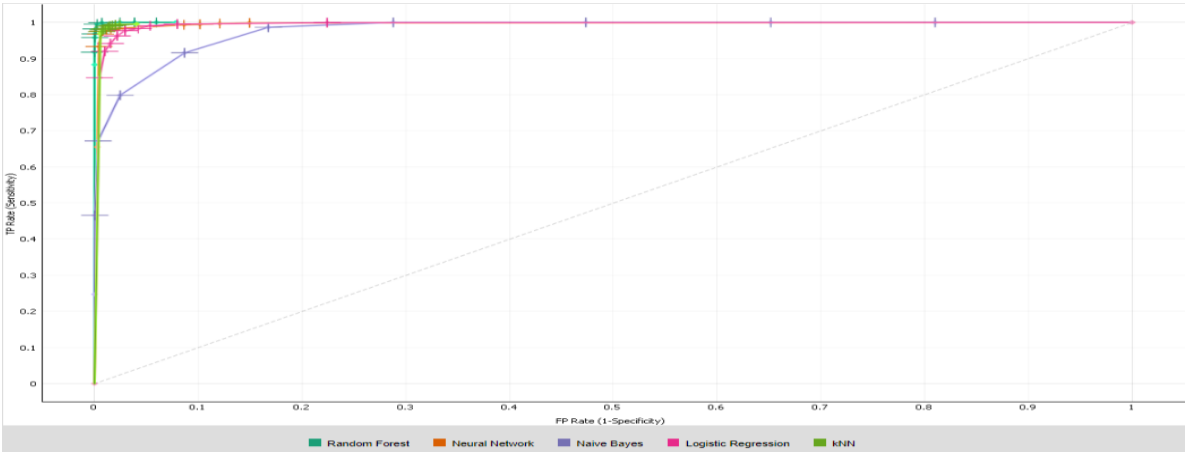


Figure 19. ROC curve by models

The results obtained were compared with 10 studies performed using the NSL-KDD database before and all findings were given in Table 21. First, in many studies on NSL-KDD, the training dataset was used as a test dataset instead of the NSL-KDD test dataset. In this context, when studies in Table 21 were analyzed in terms of the test database, it was seen that six studies used Train data for testing, and six studies (including the proposed study) used Test data. Very good accuracy rates were obtained due to training and testing on the same data set. However, this situation causes the model to memorize or to obtain low-confidence results.

Table 21. NSL-KDD comparison with literature

Research	Year	Choosing A Qualification	Test Database	Method	Accuracy Rate
Pereira et al. [59]	2012	Yes	Train	Optimum-Path Forest, Support Vector Machines, Self Organizing Maps, Bayesian Classifier	0,9661
Mohammadi et al. [60]	2012	Yes	Test	Distance Based, Neural Network Based, Decision Tree Based, MLP	0,8014
Seresht & Azmi [61]	2014	No	Train	Agent-Based Approach	0,8831
Farid et al. [62]	2014	No	Train	Decision Tree, Naïve Bayes, Supervised Classification	0,8344
Rastgeri et al. [63]	2015	Yes	Train	Genetic Algorithm Interval Rule-Based, Supervised Learning	0,7800
Singh et al. [64]	2015	Yes	Train	Online Sequential Extreme Learning Machine	0,9867
Bhattacharya et al. [65]	2015	Yes	Test	Bayes, SVM, Knn, Adabost, Cross-Validation	0,8314
Hoz et al. [66]	2015	Yes	Test	Supervised Learning	0,8800
Kang and Kim [67]	2016	Yes	Train	Feature Selection Algorithm, K-Means Clustering Algorithm	0,9693
Liu et al. [68]	2016	Yes	Test	SVM, Radial Basis Function, Neural Network, Multilayer Perceptron Neural Network	0,7460
Ozgur and Erdem [14]	2017	Yes	Test	Classifier Fusion, (Multiple Classifier Fusion Genetic Algorithms	0,9088
The Proposed Method	2023	Yes	Test	Random Forest (RF)	0,996

In addition, when Table 21 was examined, it was seen that the study with reference number [67] achieved a very low success (accuracy rate), while the study with reference number [63] had the highest performance (excluding the recommended study). When we look at the accuracy rates in general, it was possible to say that the average was between 0.85-0.90. Finally, when studies were examined in terms of the methods used, mostly Neural Networks, kNN, and SVM algorithms were used. As a result, it was seen that much higher performance was achieved when the proposed method was compared with previous studies.

5. CONCLUSION AND RECOMMENDATIONS

This study realized the detection of any abnormal behavior or attack with high accuracy performance in IoT-based network devices using ML algorithms. At this point, different models were proposed using different approaches, and their results were presented. In addition, the importance of developing cyber-physical systems in situations that threaten security and the difficulties encountered in this process was emphasized. The most up-to-date and advanced intrusion detection methods were proposed to overcome these difficulties and a comparative analysis of these methods was presented.

As a result, ML methods KNN, RF, ANN, NB, and LR algorithms were used for cyber-attack detection and the best performance was obtained with the RF algorithm. The results were compared with previous studies and it was proved that the proposed model was more successful than the others.

In addition to all this, it was revealed that cyber-attacks pose serious threats, especially in terms of infrastructure and economics, and considering that the attacks take place in environments with IoT-based systems, much larger security problems may be encountered. Therefore, intrusion detection systems should be developed to protect and prevent technological infrastructures or systems against cyber-attacks such as unauthorized access, rendering systems inaccessible. For this, an attack analysis should be made data-based and detailed.

All these results showed that artificial intelligence algorithms were an effective method for attack detection and prevention in environments where IoT devices were present. Finally, cyber security models based on intelligent algorithms need to be developed to analyze a large dataset in network-based systems. These models allow efficient and effective training and classification of large volumes of data. In addition, it was planned to make comparisons with different hybrid models in our future studies.

DECLARATION OF ETHICAL STANDARDS

The author(s) of this article declares that the materials and methods used in this study do not require ethical committee permission and/or legal-special permission.

AUTHORS' CONTRIBUTIONS

M. Hanefi CALP: Wrote the manuscript. Performed the experiments and analyzed the results.

Resul BÜTÜNER: Created the models. Performed the experiments and analyzed the results.

CONFLICT OF INTEREST

There is no conflict of interest in this study.

REFERENCES

[1] Scarfone, K., Mell P, "Guide to intrusion detection and prevention systems (IDPS)", ***NIST,*** ABD, (2007).

[2] Ganapathy, S., Kulothungan K., Muthurajkumar S.,Vijayalakshmi M., Yogesh P. & Kannan A., "Intelligent feature selection and classification techniques for intrusion detection in networks: a survey", EURASIP ***Journal on Wireless Communications and Networking***, 1:273-289, (2013).

[3] Kolias, C., Kambourakis G. & Maragoudakis M, "Swarm Intelligence in Intrusion Detection: A Survey", ***Computers and Security***, 30 (8): 625-642, (2011).

[4] Behera, S., Pradhan, A., & Dash, R. "Deep neural network architecture for anomaly based intrusion detection system". In 2018 ***5th International Conference on Signal Processing and Integrated Networks (SPIN)*** (pp. 270-274). IEEE, (2018, February).

[5] Aksu, D., & Aydin, M. A. "Detecting port scan attempts with comparative analysis of deep learning and support vector machine algorithms". ***In 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*** (pp. 77-80). IEEE, (2018, December).

[6] Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. "Deep learning approach for intelligent intrusion detection system". ***IEEE Access***, 7: 41525-41550, (2019).

[7] Hajisalem, V., Babaie, S., "A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection", ***Computer Networks,*** 136: 37-50, (2018).

[8] Inayat, Z., Gani, A., Anuar, N. B., Khan, M. K. & Anwar, S., "Intrusion response systems: Foundations, design, and challenges", ***Journal of Network and Computer Applications,*** 62: 53-74, (2016).

[9] Ashoor, A. S., Gore, S., "Difference between intrusion detection system (IDS) and intrusion prevention system (IPS)", ***In International Conference on Network Security and Applications***, 497-501, Berlin, Heidelberg, (2011).

[10] Jabez, J., Muthukumar, B., "Intrusion detection system (IDS): anomaly detection using outlier detection approach", ***Procedia Computer Science***, 48: 338-346, (2015).

[11] Quepons, I., "Vulnerability and Trust", ***PhaenEx***, 13, 2: 1-10, (2020).

[12] Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G. & Vázquez, E., "Anomaly-based network intrusion detection: Techniques, systems and challenges", ***Computers and Security***, 28: 1-2, 18-28, (2009).

[13] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization", ***in ICISSP***, Prague, Czech Republic, pp. 108-116, (2018).

[14] Ozgur, A., & Erdem, H. "Feature selection and multiple classifier fusion using genetic algorithms in intrusion detection systems", ***Journal of the Faculty of Engineering and Architecture of Gazi University***, 3(1), (2018).

[15] Demir, F. "Investigation of performance of ML methods for cyber-attack detection"", ***Journal of Balikesir University Institute of Science***, 23(2): 782-791, (2021).

[16] Gazel, S. E. R., & Bati, C. T. Determining the Best Model with Deep Neural Networks: Keras Application on Mushroom Data", ***YYU Journal of Agricultural Science***, 29(3): 406-417, (2019).

[17] Pehlivanoglu, M. K., Remzi, A. T. A. Y., & Odabas, D. E. "İki Seviyeli Hibrit Makine Öğrenmesi Yöntemi İle Saldırı Tespiti", ***Gazi Mühendislik Bilimleri Dergisi (GMBD)***, 5(3): 258-272, (2019).

[18] Cakir, B., & Angin, P. "Cyber Attack Detection Using Temporal Convolutional Networks: A Comparative Analysis". ***European Journal of Science and Technology***, 22: 204-211, (2021).

[19] Hatipoğlu, C., & Tunacan, T. Hatipoglu, C., & Tunacan, T. "Cyber Attacks and Detection Method in Turkey: A Literature Review". ***BSEU Journal of Science,*** (2021).

[20] Aytan, B., & Barisci, N. "Siber Savunma Alanında Yapay Zekâ Tabanlı Saldırı Tespiti ve Analizi". ***In Proceeding of the 2nd International Symposium on Innovative Approaches in Scientific Studies***, Samsun, (2018, December).

[21] Gurmen, C. "Performance comparison of ML methods for attack Detection systems", ***(Master's thesis, institute of science),*** (2020).

[22] Karimipour, H., Dehghantanha, A., Parizi, R. M., Choo, K. K. R., & Leung, H., "A deep and scalable unsupervised ML system for cyber-attack detection in large-scale smart grids". ***IEEE Access***, 7: 80778-80788, (2019).

[23] Kavousi-Fard, A., Su, W., & Jin, T. "A machine-learning-based cyber attack detection model for wireless sensor networks in microgrids". ***IEEE Transactions on Industrial Informatics***, 17(1): 650-658, (2020).

[24] Mousavinejad, E., Yang, F., Han, Q. L., & Vlacic, L. "A novel cyber-attack detection method in networked control systems", ***IEEE transactions on cybernetics***, 48(11): 3254-3264, (2018).

[25] AlZubi, A. A., Al-Maitah, M., & Alarifi, A. "Cyber-attack detection in healthcare using cyber-physical system and ML techniques". ***Soft Computing***, 25(18): 12319-12332, (2021).

[26] Smys, S. "DDOS attack detection in telecommunication network using ML". ***Journal of Ubiquitous Computing and Communication Technologies (UCCT)***, 1(01): 33-44, (2019).

[27] Asharf, J., Moustafa, N., Khurshid, H., Debie, E., Haider, W., & Wahab, A. "A review of intrusion detection systems using machine and deep learning in internet of things: challenges, solutions and future directions". ***Electronics***, 9(7): 1177, (2020).

[28] Rashid, M. M., Kamruzzaman, J., Hassan, M. M., Imam, T., & Gordon, S. "Cyberattacks detection in IoT-based smart city applications using ML techniques". ***International Journal of environmental research and public health***, 17(24): 9347, (2020).

[29] Alsamiri, J., & Alsubhi, K. "Internet of things cyber-attacks detection using ML". Int. J. Adv. Comput. Sci. Appl, 10(12): 627-634, (2019).

[30] Dutta, V., Choraś, M., Pawlicki, M., & Kozik, R. "A deep learning ensemble for network anomaly and cyber-attack detection". ***Sensors***, 20(16): 4583, (2020).

[31] Awan, M. J., Farooq, U., Babar, H. M. A., Yasin, A., Nobanee, H., Hussain, M., ... & Zain, A. M. "Real-time DDoS attack detection system using big data approach". ***Sustainability***, 13(19): 10743, (2021).

[32] Wu, M., Song, Z., & Moon, Y. B. "Detecting cyber-physical attacks in CyberManufacturing systems with ML methods". ***Journal of intelligent manufacturing***, 30(3): 1111-1123, (2019).

[33] Savaş, T. & Savaş, S. "Tekdüzen Kaynak Bulucu Yoluyla Kimlik Avı Tespiti için Makine Öğrenmesi Algoritmalarının Özellik Tabanlı Performans Karşılaştırması". ***Politeknik Dergisi*** , 25 (3): 1261-1270 . DOI: 10.2339/politeknik.1035286, (2022).

[34] Catania C.A., Garino C.G., "Automatic network intrusion detection: Current techniques and open issues", ***Computers & Electrical Engineering***, 38 (5): 1062-1072, (2012).

[35] Hubballi N., Suryanarayanan V., "False alarm minimization techniques in signature-based intrusion detection systems: A survey", ***Computer Communications***, 49: 1-17, (2014).

[36] Cunningham R.K., Lippmann R.P., Fried D.J., Garfinkel S.L., Graf I. , Kendall K., Wyschogrod D. & Zissman M.A., "Evaluating intrusion detection systems without attacking your friends: The 1998 DARPA intrusion detection evaluation", (1999).

[37] Tavallaee M., Bagheri E., Lu W. & Ghorbani A.A., "A detailed analysis of the KDD CUP 99 dataset", ***Proceedings of the Second IEEE international conference on Computational intelligence for security and defense applications***, Piscataway, NJ, USA, (2009).

[38] ***NSL-KDD***, "Download Link of NSL-KDD in Github". https://github.com/ati-ozgur/NSL_KDD. January 17, (2017). Access Date: 02 February 2022.

[39] Özgür A., Erdem H., "A Review of KDD99 Dataset Usage in Intrusion Detection and ML between 2010 and 2015", ***PeerJ Preprints*** 4:e1954v1, (2016).

[40] Marquez-Viloria, D., Castano-Londono, L., & Guerrero-Gonzalez, N., "A Modified KNN Algorithm for High-Performance Computing on FPGA of Real-Time m-QAM Demodulators". ***Electronics***, 10(5): 627., (2021).

[41] Rottondi, C. Barletta, L. Giusti, A. Tornatore, M. "Machine-Learning Method for Quality of Transmission Prediction of Unestablished Lightpaths". ***J. Opt. Commun. Netw.***, 10: A286–A297, (2018).

[42] Pérez, A.E., Torres, J.J.G., González, N.G. "KNN-based Demodulation in gridless Nyquist-WDM Systems affected by Interchannel Interference". ***In Proceedings of the OSA Advanced Photonics Congress (AP)*** 2019 (IPR, Networks, NOMA, SPPCom, PVLED), Burlingame, CA, USA, 29 July–1 August 2019; p. SpTh1E.3, (2019).

[43] Han, J., Pei, J., Kamber, M. "Data mining: concepts and techniques. Massachusetts", ***USA: Morgan Kaufmann Publishers***. 978-0-12-381479-1, (2011).

[44] Breiman, L., "Random Forests", ML, ***Kluwer Academic Publishers***, 45(1): 5-32, (2001).

[45] Resende, P. A. A., & Drummond, A. C. "A survey of random forest based methods for intrusion detection systems". ***ACM Computing Surveys (CSUR)***, 51(3): 1-36, (2018).

[46] Akar, O., Gungor, O., "Classification of multispectral images using Random Forest algorithm", ***Journal of Geodesy and Geoinformation***. 1 (2): 139-146. DOI: 10.9733/jgg.241212.1t, (2012).

[47] Archer, K.J., "Empirical Characterization of Random Forest Variable Importance Measure, Computational Statistical Data Analysis", ***Computational Statistics & Data Analysis***, 52(4): 2249-2260, (2008).

[48] Calp, M. H., & Kose, U. "Estimation of burned areas in forest fires using artificial neural networks". ***Ingeniería Solidaria***, 16(3): 1-22, (2020).

[49] Calp, M. H. "An estimation of personnel food demand quantity for businesses by using artificial neural networks", ***Journal of Polytechnic***, 22(3): 675-686, (2019).

[50] Bayram, S., Kaplan, K., Kuncan, M., Ertunç H. M.. "Ball Bearings space of time Statistical Feature Extraction and Neural Networks with Error Estimation Method Size", ***Automatic Control National Meeting***, TOK2013, Malatya, 26-28 September, (2013).

[51] Öztemel, E. "Yapay sinir ağları", ***PapatyaYayincilik***, Istanbul, (2003).

[52] Deng, H., Sun, Y., Chang, Y., Han, J., "Probabilistic Models for Classification". C.C. Aggarwal (Eds.), Data Classification Algorithms and Applications (pp. 67-70), ***CRC Press***, New York, USA, (2015).

[53] Bayes, T., LII. "An essay towards solving a problem in the doctrine of chances". By the late Rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFR S. ***Philosophical transactions of the Royal Society of London***, 1763(53): 370-418, (1958).

[54] Yildiz, H.K., et al. "A new feature extraction method for text classification". ***in 2007 IEEE 15th Signal Processing and Communications Applications***. June 2007. Eskisehir, Turkey: IEEE. DOI: 10.1109/SIU.2007.4298870, (2007).

[55] Hosmer, D. W., Lemeshow, S., "Applied Logistic Regression", ***John Wiley & Sons***, New York, 5-50 (1989).

[56] Kleinbaum, G., D., "A Self-learning Text Logistic Regression", ***Springer***, Atlanta, (1994).

[57] Kaya, Y., "Predictive modeling in motor caravan insurance and comparison of methods applied", ***(Master's thesis)***, Graduate School of Natural and Applied Sciences, Beykent University, Istanbul, (2017).

[58] Tunç, Ü., Atalar, E., Gargi, M. S., Ergül Aydin, Z. "Classification of Fake, Bot, and Real Accounts on Instagram Using Machine Learning". ***Politeknik Dergisi,*** 1-1. https://doi.org/10.2339/politeknik.1136226, (2023).

[59] Pereira C.R., Nakamura R.Y.M., K., Costa A.P. & Papa J.P., "An Optimum-Path Forest framework for intrusion detection in computer networks", ***Engineering Applications of Artificial Intelligence***, 25: 1226-1234, (2012).

[60] Mohammadi M., Raahemi B., Akbari A. & Nassersharif B., "New class-dependent feature transformation for intrusion detection systems", ***Security and Communication Networks,*** 5: 1296-1311, (2012).

[61] Seresht N.A., Azmi R., "MAIS-IDS: A distributed intrusion detection system using multi-agent AIS approach", ***Engineering Applications of Artificial Intelligence***, 35: 286-298, (2014).

[62] Farid D.M., Zhang L., Rahman C.M., Hossain M.A. & Strachan R., "Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks", ***Expert Systems with Applications***, 41: 1937-1946, (2014).

[63] Rastegari S., Hingston P. & Lam C.P., "Evolving statistical rulesets for network intrusion detection", ***Applied Soft Computing***, 33: 348-359, (2015).

[64] Singh R., Kumar H. & Singla R. K., "An intrusion detection system using network traffic profiling and online sequential extreme learning machine", ***Expert Systems with Applications***, 42: 8609-8624, (2015).

[65] Bhattacharya S., Selvakumar S., "LAWRA: a layered wrapper feature selection approach for network attack detection", ***Security and Communication Networks***, 8: 3459-3468, (2015).

[66] Hoz L.E.D., Ortiz A., Ortega J. & Prieto B., "PCA filtering and probabilistic SOM for network intrusion detection", ***Neurocomputing***, 164: 71-81, (2015).

[67] Kang S.H., Kim K.J., "A feature selection approach to find optimal feature subsets for the network intrusion detection system", ***Cluster Computing***, 19: 325-333, (2016).

[68] Liu Q., Yin J., Leung V.C.M., Zhai J.H., Cai Z. & Lin J., "Applying a new localized generalization error model to design neural networks trained with extreme learning machine", ***Neural Computing and Applications***, 27: 59-66, (2016).