Bitlis Eren Üniversitesi Fen Bilimleri Dergisi
BİTLİS EREN UNIVERSITY JOURNAL OF SCIENCE
ISSN: 2147-3129/e-ISSN: 2147-3188
VOLUME: 12 NO: 2 PAGE: 465-477 YEAR: 2023
DOI:10.17798/bitlisfen.1240469

# Analysis of Intrusion Detection Systems in UNSW-NB15 and NSL-KDD Datasets with Machine Learning Algorithms

Fuat TÜRK[1*]

[1] *Cankiri Karatekin University, Department of Computer Engineering, Cankiri/Turkey (ORCID: 0000-0001-8159-360X)*

**Keywords:** Instruction Detection Systems, Network Attacks, NSL-KDD Dataset, UNSW-NB15 Dataset.

**Abstract**

The use of intelligent devices in almost every sector, and the provision of services by private and public institutions through network servers, cloud technologies, and database systems are now mostly remotely controlled. Due to the increasing demands on network systems, unfortunately, both malicious software and users are showing more interest in these areas. Some organizations are facing almost hundreds or even thousands of network attacks daily. Therefore, it is not enough to solve the attacks with a virus program or a firewall. Detection and accurate analysis of network attacks are crucial for the operation of the entire system. With the use of deep learning and machine learning, attack detection, and classification can be successfully performed. This study conducted a comprehensive attack detection process on the UNSW-NB15 and NSL-KDD datasets using existing machine learning and deep learning algorithms. In the UNSW-NB15 dataset, an accuracy of 98.6% and 98.3% was achieved for two-class and multi-class classification, respectively, and 97.8% and 93.4% accuracy were obtained in the NSL-KDD dataset. The results prove that machine learning algorithms are an effective solution for intrusion detection systems.

## 1. Introduction

Today, the rapid development of big data, cloud technologies, and smart devices has significantly increased our dependence on internet systems. In addition, the use of the internet in economic, military, and institutional contexts has become extremely important. For this reason, data confidentiality, data integrity, and information security are fundamental tasks. While network authorities try to meet the increasing needs and security demands, malicious software and intruders, on the other hand, try to infiltrate systems, and destroy and change information. This situation has advanced so much that intrusion detection systems and intrusions now reach the level of interstate cyber wars. Therefore, network systems should be developed in terms of confidentiality, integrity, and usability, and information security should be prioritized. Intrusion detection systems (IDS) are one of the biggest problems caused by malicious users in cybersecurity [1].

IDS is mainly used to detect suspicious logins. It can be in the form of software or hardware or a combination of both. Two methods are mostly preferred in IDS screening. The first is the HIDS and the second is the NIDS. HIDS follows the network interfaces and configurations of the target machine and requires certain settings compatible with the server [2], [3]. With the proliferation of attacks, databases are forced to constantly update. Also, specification-based types require expert experience to detect intrusions. Artificial intelligence can be used easily since the detection of such anomaly situations is a classification problem. For this reason, many data sets have been created to control intrusion detection systems [4].

The main ones are NSL-KDD, UNSW-NB15, KDDCUP99, and CICICS2017. Existing machine learning algorithms are used with these

*Corresponding author: turk_fuat@hotmail.com
*Received: 22.01.2023, Accepted: 24.04.2023*

datasets and offer extremely important ideas for intrusion detection systems.

It is possible to detect features and select appropriate features using machine learning algorithms, filter methods, or learning-based methods. Additionally, ensemble learning methods are also in demand for feature selection. Ensemble algorithms in machine learning are a technique that aims to achieve higher performance by combining multiple learning algorithms. These algorithms can combine the predictions of various learning algorithms to obtain more accurate results. Therefore, ensemble algorithms can be used to eliminate the weak points of individual algorithms and make more general predictions. Using a combination of machine learning and deep learning algorithms sometimes outperforms classification problems [5]. At this stage, it is extremely important to determine which models will be used and the strengths and weaknesses of the models. For this reason, existing machine learning or deep learning algorithms should be applied to accepted data sets. This study involved an extensive detection analysis of the NSLKDD and UNSW-NB15 datasets. Consequently, it is crucial to offer new approaches to intrusion detection systems and to develop different solutions. The workflow of the proposed approach model is shown in Figure 1.

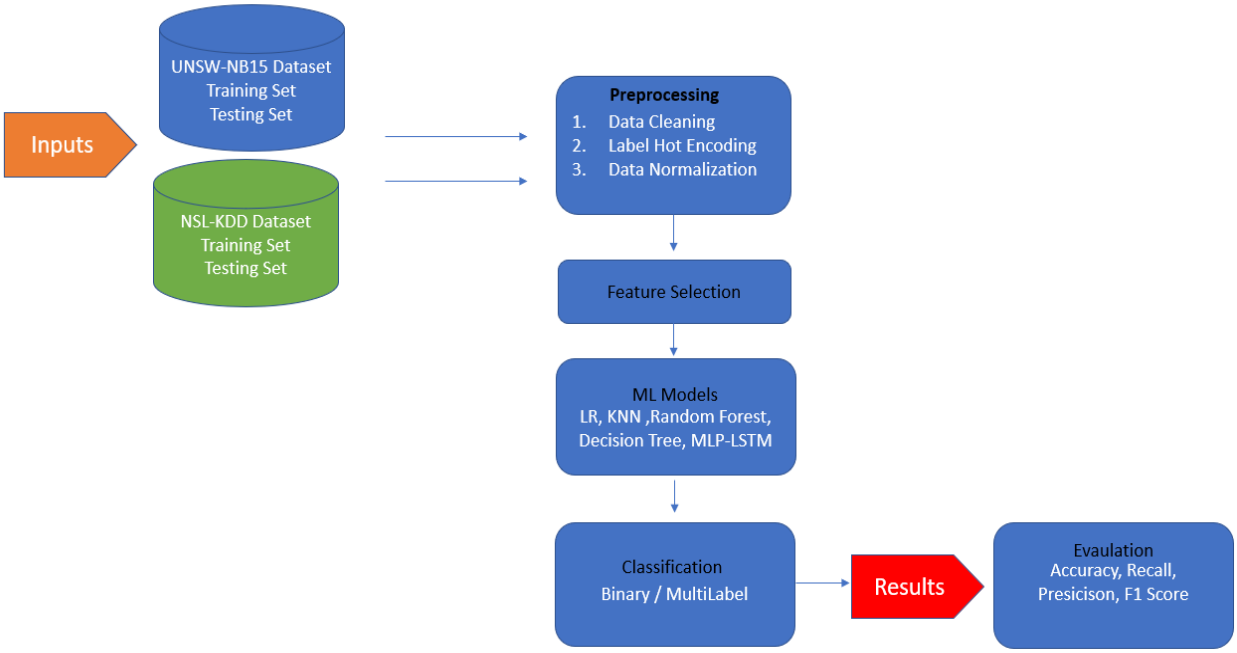


**Figure 1.** Workflow diagram of the proposed system.

According to the proposed workflow line, data cleaning is performed before all the algorithms are applied. To fix data cleaning dataset defects, follow the steps of concatenating the textual values, processing the empty columns according to their nature, and converting the values stored as text type to number type. One hot encoding stage is the step of converting nominal properties to numeric values before machine learning. The data normalization process is considered a process where the attribute values are scaled in the range of [0 and 1] and the computational load is reduced. Feature selection is an important step in artificial intelligence model building. First, it eliminates the dimensionality problem caused by having many features. Second, it saves the model from workload with many features and turns it into a simple and openable structure. Therefore, it is wise to simplify the model with the effect of increasing complexity and training/testing time. After the feature selection process, machine learning algorithms, Multi-Layer Perception, and Long-Short Term Memory algorithms are used for classification. The classification was carried out as both binary and multiclass. Finally, experimental results were compared in terms of evaluation criteria.

## 2. Literature Review

Geurts et al. conducted research on the intrusion detection system with the Bi-Directional LSTM model. In this study, they explained that intrusion detection systems are a basic layer incorporated into the network system. They stated that due to the excessive amount of data traffic on the network, attackers could cause great harm to the network and its users. The Bidirectional LSTM model gave results

with 99% accuracy for both datasets. Paragraphs following the first paragraph should begin with the paragraph indentation [5]. Basati and Faghih proposed an architecture PDAE for the security of IoT devices against network attacks. In this study, they stated that due to the limited resources of IoT devices, a high-fidelity neural network with a lightweight and efficient architecture is needed for intrusion detection. For this reason, they stated that the traditional architectural structures of neural networks were not feasible. The proposed PDAE greatly reduces the number of parameters, the amount of memory, and the need for processing power, while increasing the accuracy of the model. The results were calculated as superior to the existing algorithms in terms of both accuracy and performance [6]. Cil et al. conducted a study on the importance of early detection of network traffic in the fight against network attacks. The proposed model is a deep network that detects attacks on packet instances. As a result of the experiments, attack types were determined with an accuracy rate of 94.57% [7]. Amaizu et al. proposed a unified and efficient network attack detection framework for B5G networks. The proposed model includes multi-layered detectors combined with the feature extraction algorithm and was created to detect the r network attacks as well as revert the DDoS attack type. The results showed that the framework could detect network attacks with a high accuracy score of 99.66% [8]. Gowthul et al. proposed an SVM-based DEHO Classifier model to detect DDoS attacks. The main purpose of this article is to ensure that they are best detected as normal data samples and malicious/hacked data samples. The proposed approach was examined for four different databases. Experimental results reveal that the performance of detection system using this approach is higher than that ofother approaches [9]. Mushtaq et al. proposed a two-stage auto-encoder-based LSTM architecture. Experimental results showed that the proposed AE-LSTM performance has fewer prediction errors compared to other deep and shallow machine learning techniques. The NSL-KDD dataset showed 89% classification accuracy [10]. Choudhary and Kesswani proposed a deep learning-based model for the detection of unauthorized attacks. They mentioned that the application of IoT technology is increasing rapidly, resulting in the need for the most efficient model to detect malicious activities as quickly and accurately as possible, and Deep Neural Networks are used to identify attacks. The performance of DNN to accurately identify the attack was evaluated on the most used datasets. Experimental results showed that the accuracy rate of the proposed method using DNN is over 90% [11].

Serinelli et al. proposed An Intrusion Detection System Architecture ANIDINR (an anomaly-based NIDS in R). In this study, they stated that the protection of computer networks is one of the most important and difficult problems in cyber security. The main purpose here is to try to provide step-by-step guidance on methodology selection and execution for training Machine and Deep Learning models. There is also a focus on developing ANIDINR to overcome the problems of detection of well-known attacks and the complex and up-to-date collection of rules. Based on this setup, the proposed system yielded over 90% accuracy results on the two datasets (NSL-KDD and KDDCup 1999) [3]. Moualla et al. proposed a machine learning-based system for the intrusion detection system. The proposed system is a dynamically scalable multi-class machine learning-based network IDS. The outputs of the extreme learning machine classifier are used as the inputs of a fully connected layer followed by a logistic regression layer to make smooth decisions for all classes. The results show that it outperforms the related studies in terms of accuracy [12]. Mohammadpour et al. proposed a convolutional neural network-based system for the intrusion detection system. In this paper, a deep learning method is proposed to implement an effective and flexible NIDS. The model was run with the NSL-KDD' dataset, which is a benchmark dataset for network intrusion. Experimental results of a 99.79% detection rate were obtained when compared with the test dataset. In line with the results, they stated that CNNs can be applied as a learning method for IDS. Paragraphs following the first paragraph should begin with the paragraph indentation [13].

Apart from these studies, machine learning and ensemble learning algorithms have been used in different fields. Ayşe et. al. proposed a new super Community learning model to enable early diagnosis of diabetes mellitus. The proposed super-learner model was created as a result of a case study with four basic learners (logistic regression, decision tree, random forest, and gradient boosting) and a meta-learner (support vector machines). This model found the early-stage diabetes risk estimation to be 99.6%, 92%, and 98%, respectively, in three different datasets [14]. In a study in the agricultural sector, Buyrukoğlu proposed a new hybrid model to predict the presence of Salmonella in agricultural surface waters based on a combination of heterogeneous ensemble approach for feature selection, clustering, regression, and classification algorithms. The ensemble ANN+ RF model achieved the highest performance and performed well, with a prediction accuracy of 94.9% [15]. In his work on finance,

Buyrukoğlu aimed to analyze promising cryptocurrencies with deep learning methods. Five promising cryptocurrencies were analyzed using LSTM communities and single-base LSTM networks. The results of the study revealed that LSTM network ensembles do not always provide better accuracy performance than single-base LSTM network [16].

## 3. Materials and Method

### 3.1. Dataset

**UNSW-NB15 Dataset:** The UNSW-NB15 dataset was created in the Cyber Lab of the Australian Cyber Security Center. The main objective of the dataset is to obtain a combination of real regular activities and synthetic modern attack behavior. The dataset consists of approximately 2 million records with 49 different features extracted using some special algorithms. The data set can be divided into normal and abnormal. However, it includes nine types of attacks [17],[18]. Figure 2 shows the distribution of the data set separately.

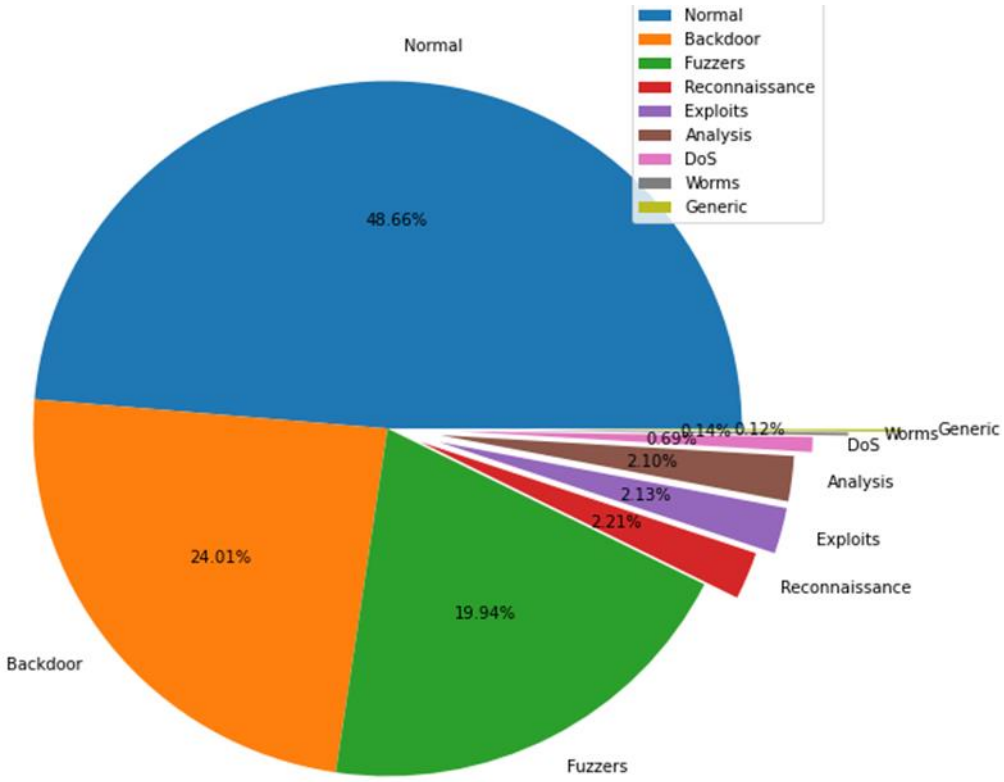


**Figure 2.** Pie chart distribution of multi-class labels for UNSW-NB15.

**Fuzzer attack:** These are the types of attacks obtained with randomly generated data trying to hack the program or network.

**Analysis:** Hosts different types of attacks, including web scripts for port scanning and spam-like email.

**Backdoor:** Backdoor is a technique where attackers use a legitimate system portal to gain illegal access.

**Denial of service (DOS):** These are the types of attacks in which the server or network is busy so that the users of the system cannot access it and cause it to interfere with the services of the host on the Internet.

**Exploits:** Attacks that take advantage of a vulnerability caused by any bug and attempt to disrupt trusted behavior on the network.

**Generic:** This analysis can be applied to block verification code passwords, broadcast, and send messages.

**Reconnaissance:** These are the types of attacks that gather preliminary information about any public network or target host. Based on the collected

information, it is used to infiltrate target hosts or networks [19].

**Shell Code:** Shell code is an attack method that uses code to exploit a software vulnerability.

**Worm:** These are the types of attacks that regenerate and increase themselves to start on one computer and spread to another [20].

**NSL-KDD Dataset:** The NDSL-KDD dataset is derived as a new dataset, consisting of records determined from the complete KDD dataset, which poses no problem in correcting the errors in the KDD-99 cup dataset [21]. However, the dataset is subject to certain problems, such as not being representative of low-footprint attacks [22],[23]. The NSL-KDD dataset has better reduction rates and no duplicate records in the test set. Because NSL-KDD has fewer data points than KDD-99, it is inexpensive in terms of workload to use in training machine learning models. It can be classified as normal and abnormal, as in the UNSW-NB15 dataset. In addition, it is possible to multiclass as normal, DOS, R2L, U2R, and probe. Figure 3 shows the distribution of the data set separately.

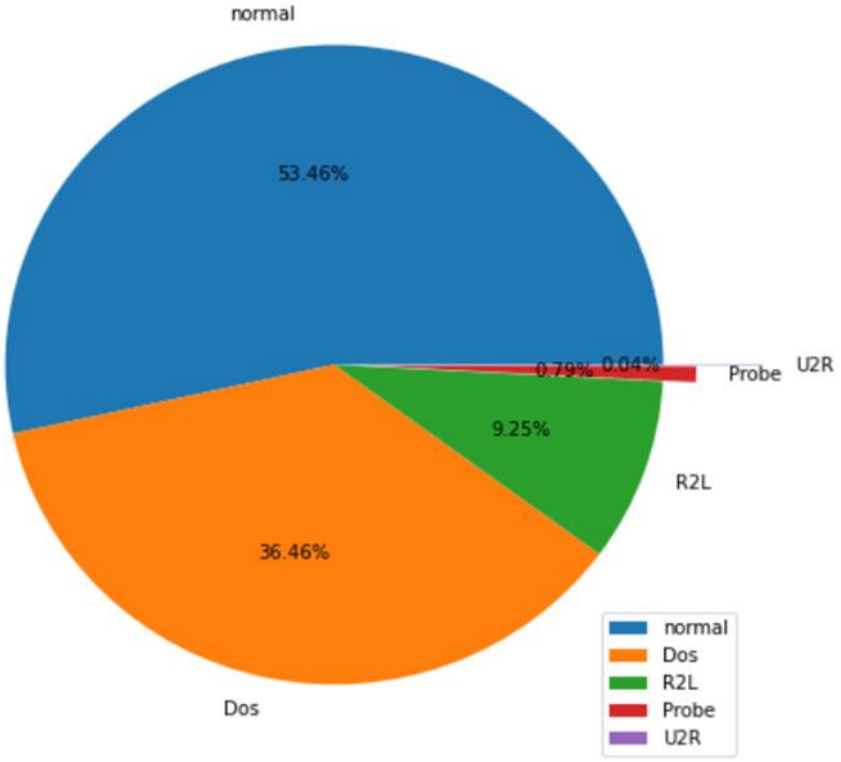


**Figure 3.** Pie chart distribution of multi-class labels for NSL-KDD.

**Denial of Service (DOS):** It is a type of attack that consumes the resources of the other party and thus renders it unable to meet requests.

**Remote to Local (R2L):** It is a type of attack that intrudes on another machine from a remote machine and gains local access to that machine.

**User-to-Root (U2R):** This is an unauthorized access method for root privileges. It is a form of attack that can enter a normal account into the system to be accessed, but tries to gain root / administrator privileges due to some security vulnerabilities in the system.

**Probing:** We can summarize it as the purpose of surveillance and other research attacks. The main purpose is to gather information about the remote machine.

### 3.2. Machine Learning Algorithms Used

**Logistic Regression:** Logistic Regression (LR) is a machine learning method for creating the most appropriate model to establish a relationship between class variables and features. Often, in binary class problems (with 0 and 1), the probability of being included in the class for an observation produces a value between (1) and non-existence (0). However, it can be adapted for multi-class problems with simple adjustments [24].

**K-Nearest Neighbors**: K-Nearest Neighbors (KNN) is a widely used classification algorithm. It is preferred in many classification problems due to its easy interpretation and low computation time. The

selection of the k parameter is extremely important in the KNN algorithm [25].

**Random Forest (RF):** Random Forest is an ensemble learning classification and regression algorithm suitable for grouping data into classes. During the training phase, a series of decision trees are created that are then used for class prediction. In the calculation process, the classes of all individual trees are considered, and the class with the highest votes is considered as the output [26].

**Decision Tree:** The Decision Tree has a root node, branches, and leaf nodes. Testing an attribute is in each internal node, and the result of the test is in the branch and class tags. The root node is the top node in the tree. A decision tree is a tree in which each node represents a feature, each link represents a decision, and each leaf represents a result [27].

### 3.3. Deep Learning Architecture Used

**Multi-Layer Perception:** Multi-Layer Perception (MLP) neural networks contain units arranged in layers in their internal structure. These units are the input layer, one or more hidden layers, and the output layer.

The input layer transfers the input to the next layers. Hidden volume nodes have non-linear enabled functionality, and outputs are linearly enabled. For true three-layer MLP, all inputs are also directly connected to all outputs [28]. Figure 4 shows the MLP neural network architecture.

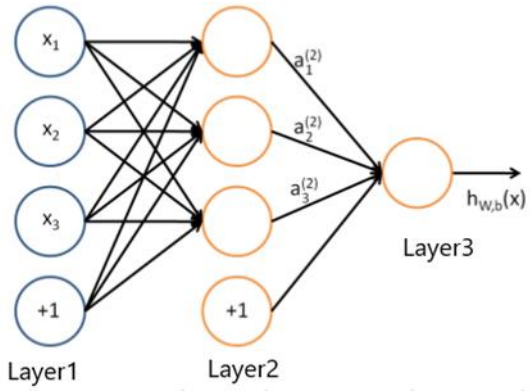


**Figure 4.** MLP neural network architecture.

**Long-Short-Term Memory:** LSTM was designed to overcome these error backflow problems. Although it is mentioned together with deep learning algorithms, it should be considered as a sub-unit of machine learning methods. In the noisy state, even with compact input arrays, it can learn to bridge time intervals over 1000 steps without losing its short time delay capabilities [29]. An efficient, gradient-based algorithm achieves this for an architecture that enforces a constant stream of errors that does not explode or disappear through each unit's internal states. In principle, an LSTM could use memory cells to remember long-range information and monitor various attributes of the text it is currently processing. An LSTM unit consists of a cell, an entry gate, an exit gate, and a forgotten gate [30]. The cell gate can be expressed as the memory of the network that carries the information across the cells for prediction purposes. The input gate executes the function of updating the cell state. It decides whether to update the information according to the sigmoid function operation. The exit gate decides what the next cell's entrance will be. It is also used in forecasting. The forget gate is the gate that decides what information to forget or keep [31], [32]. The figure shows the LSTM architectural structure. Figure 5 shows the LSTM architectural structure.

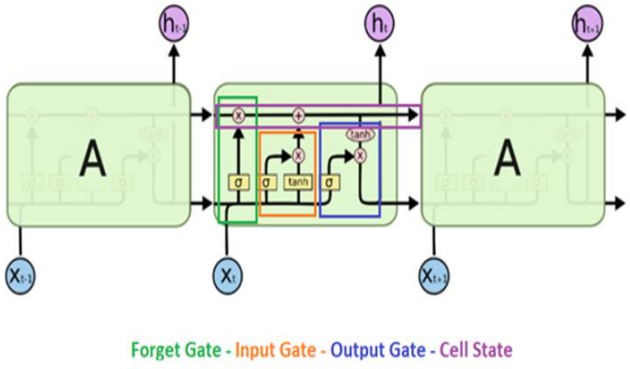


**Figure 5.** LSTM network architecture.

### 3.4. Performance Evaluation Metrics

Unless the images within the classes in the dataset are balanced, the measurement of classification accuracy is not sufficient on its own and may give deceptive results. Performance indicators for each class in the dataset are calculated based on the confusion matrix. These indicators are Accuracy, Recall, Precision, and F-1 Score values. The explanations of these parameters are given below in Table 1:

**Table 1.** Calculation criteria for evaluation metrics.

| Total Instances | Predicted No | Predicted Yes | |
|---|---|---|---|
| Actual No | TN (True Negative) | FP (False Positive) | |
| Actual Yes | FN (False Negative) | TP (True Positive) | **Recall** |

| | Precision | Accuracy |
|---|---|---|

Accuracy=TP/(Total Instances) (1)

Recall=TP/(Total Actual Yes) (2)

Precision=TP/(Total Predicted Yes) (3)

F1score=(2*Prec*Recall)/(Prec+ Recall) (4)

## 4. Results and Discussion

In both data sets, a random distribution phase was applied at 80% and 20% for the training and testing phases, respectively. In the UNSW-NB15 dataset selected as an input, 5 basic features were extracted, and for the NSL-KDD dataset, 9 basic features were extracted. In addition, the Pearson Correlation method was used for feature selection. The training process of the datasets was carried out using the GTX1050 TI graphics card and the TensorFlow-GPU 2.3 library.

Before the models were trained, certain hyperparameter settings were made for the classification algorithms. The number of trees for the random forest algorithm is set to 120. The learning rate was taken as 0.01. The logistic regression iteration number was determined as 100 random states=0, decision tree max-leaf node=default, LSTM activation function was determined as tanh.

### 4.1 UNSW-NB15 Training Phase

Table 2 shows the evaluation results for UNSW-NB15. As can be seen from the results, the highest accuracy value for the binary class was calculated with the Random Forest and for the multi-class Logistic Regression algorithm. Their success in classification problems and their structures that provide good predictions show that Random Forest and Logistic Recession algorithms give good results. Since the Random Forest algorithm is based on Ensemble Learning, it is usually high in classification problems. In addition, it gives higher success rates on the definition of Logistic Regression in multi-class problems. In addition, we can say that the hyperparameters are chosen correctly.

**Table 2.** Evaluation results for UNSW-NB15.

| | **Binary Class** | | | | | **Multi Class** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Accuracy | Recall | Precision | F1-Score | Total Time | Accuracy. | Recall | Precision | F1-Score | Total Time |
| LR | 0.978 | 0.96 | 0.98 | 0.97 | 3.1 | **0.983** | **0.97** | **0.98** | **0.98** | 4.2 |
| KNN | 0.984 | 0.97 | 0.98 | 0.98 | 10.6 | 0.975 | 0.98 | 0.87 | 0.98 | 11.5 |
| Random Forest | **0.986** | **0.98** | **0.98** | **0.98** | 2.5 | 0.976 | 0.98 | 0.89 | 0.98 | 2.9 |
| Decision Tree | 0.980 | 0.98 | 0.97 | 0.97 | **2.3** | 0.973 | 0.98 | 0.87 | 0.90 | **2.7** |
| MLP | 0.983 | 0.97 | 0.98 | 0.98 | 12.4 | 0.975 | 0.97 | 0.88 | 0.91 | 14.4 |
| LSTM | 0.976 | 0.96 | 0.98 | 0.97 | 14.5 | 0.971 | 0.98 | 0.85 | 0.90 | 16.1 |

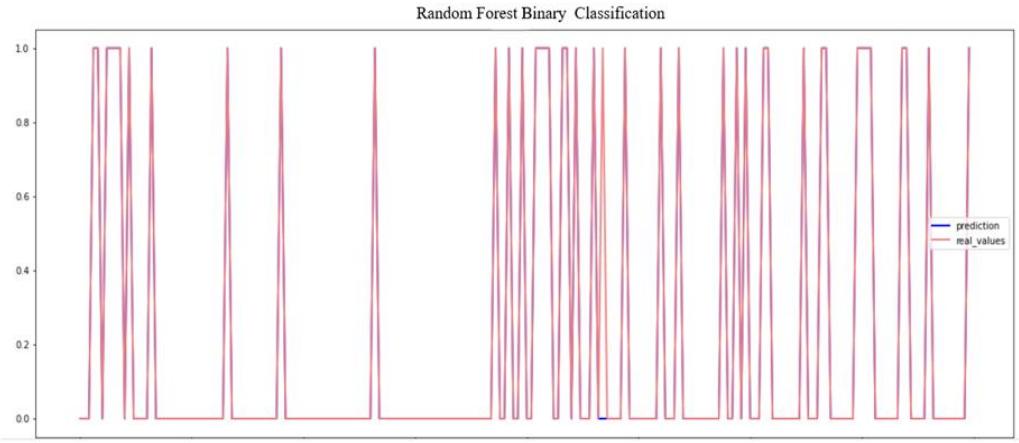


**Figure 6.** Random Forest binary classification prediction data signal.

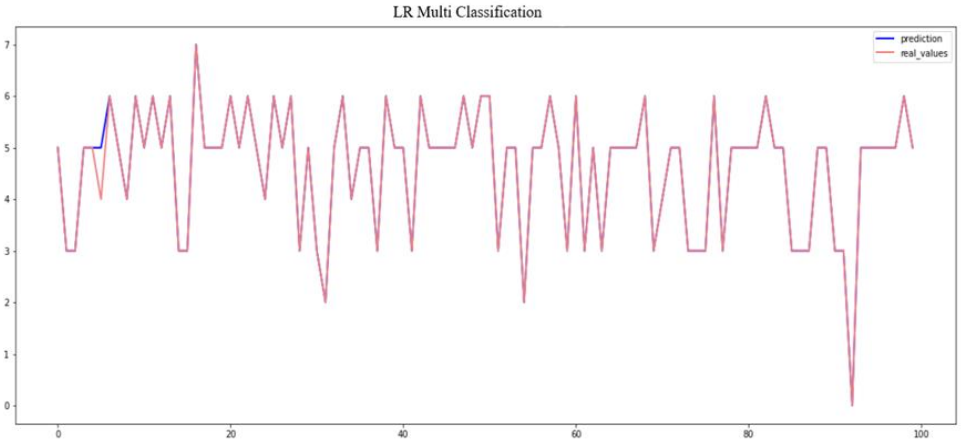


**Figure 7.** LR multi-label classification prediction data signal.

Figures 8 and 9 show the accuracy graphs of the RF and LR models. We can say that the results are close to each other and successful.

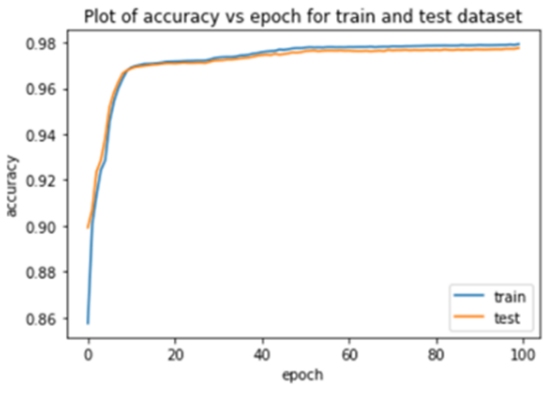


**Figure 8.** RF accuracy graph for binary class.

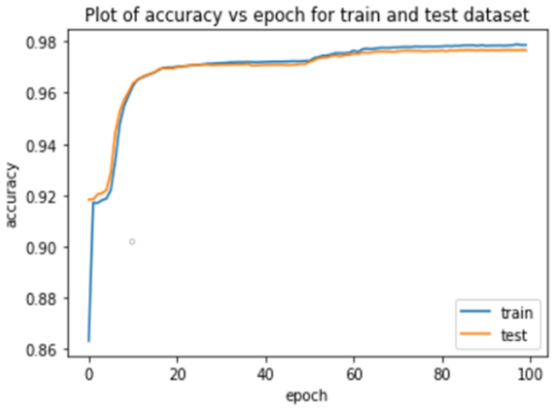


**Figure 9.** LR accuracy graph for multi-label class.

### 4.2 NSL-KDD Training Phase

Table 3 shows the evaluation results of NSL-KDD. As can be seen from the results, the highest accuracy value for the binary class was calculated using the MLP algorithm and for the multi-class LSTM algorithm.

**Table 3.** Evaluation results for NSL-KDD.

| | **Binary Class** | | | | | **Multi Class** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Accuracy | Recall | Precision | F1-Score | Total Time | Accuracy. | Recall | Precision | F1-Score | Total Time |
| LR | 0.966 | 0.97 | 0.97 | 0.97 | **1.9** | 0.878 | 0.88 | 0.86 | 0.89 | 3.3 |
| KNN | 0.955 | 0.95 | 0.96 | 0.96 | 12.5 | 0.928 | 0.92 | 0.91 | 0.91 | 14.6 |
| Random Forest | 0.957 | 0.96 | 0.96 | 0.96 | 2.1 | 0.913 | 0.91 | 0.90 | 0.90 | **3.1** |
| Decision Tree | 0.967 | 0.97 | 0.97 | 0.97 | 2.3 | 0.905 | 0.89 | 0.90 | 0.90 | 3.7 |
| MLP | **0.978** | **0.98** | **0.97** | **0.98** | 11.7 | 0.892 | 0.88 | 0.89 | 0.89 | 15.1 |
| LSTM | 0.975 | 0.97 | 0.97 | 0.97 | 16.3 | **0.934** | **0.93** | **0.94** | **0.93** | 17.2 |

Analysis of data signals because of dual-class and multi-class testing is shown in Figures 10 and 11. Considering the estimated signal and actual signals, there is some loss in the initial and intermediate stages. However, this loss is at an acceptable level.

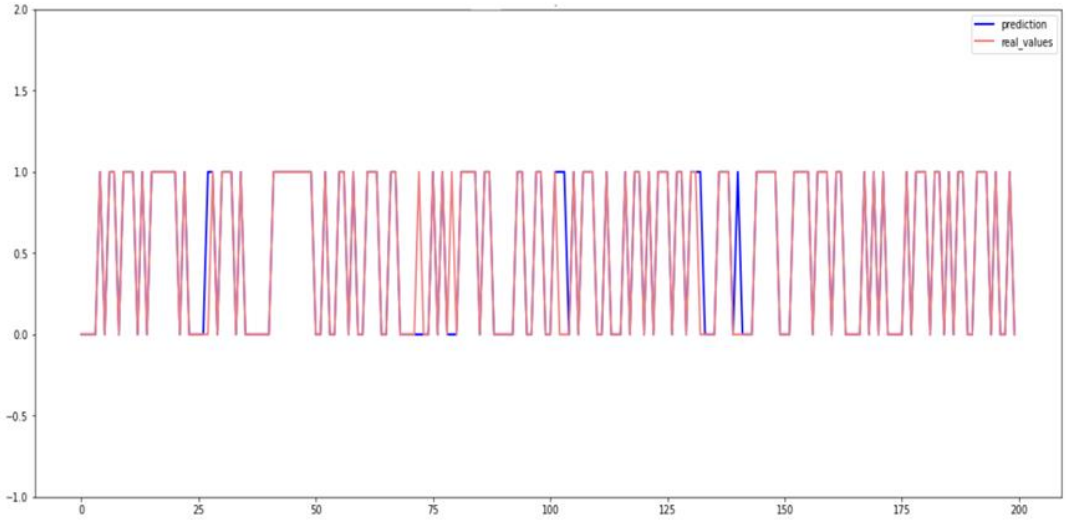


**Figure 10.** MLP binary classification prediction data signal.

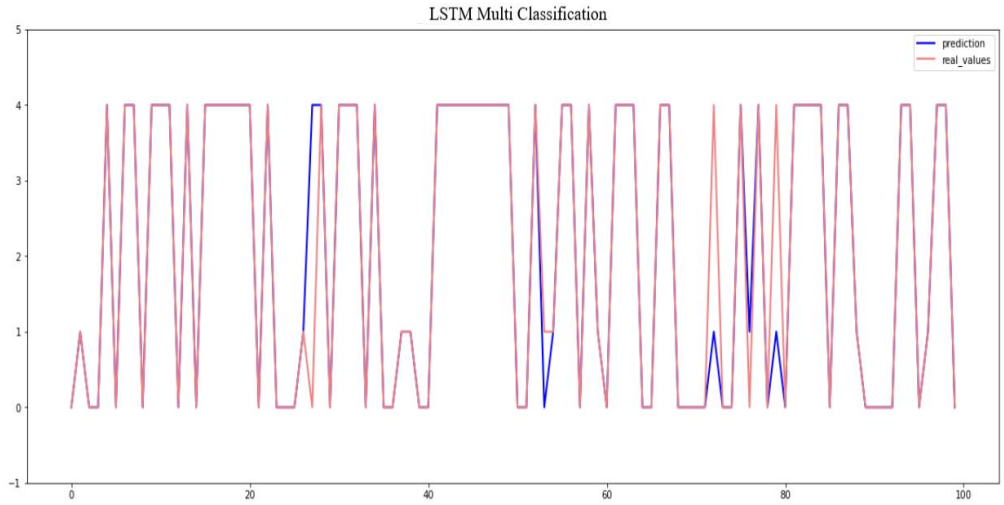


**Figure 11.** LSTM multi-label classification prediction data signal.

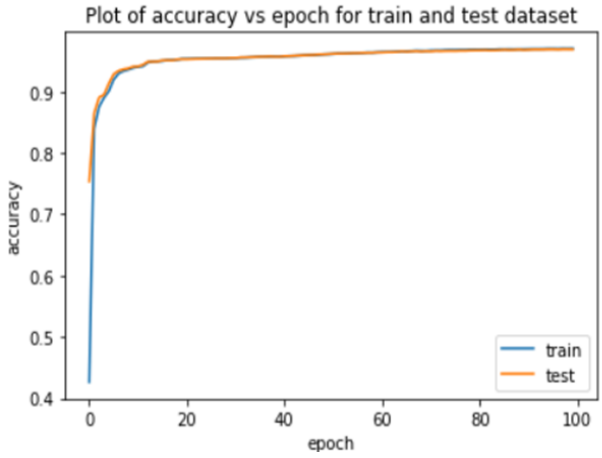


**Figure 12.** MLP accuracy graph for binary class.

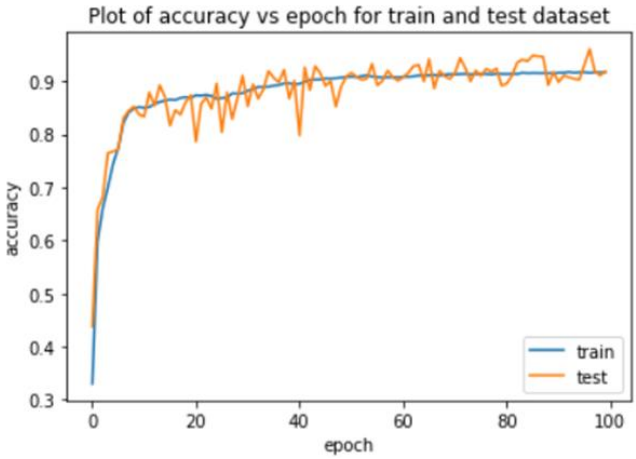


**Figure 13.** LSTM accuracy graph for binary class.

Table 4 shows a comparison of some studies in the literature and the methods we recommend. When the results are examined, the success of the Random Forest model in classifying the data for each study is remarkable. In addition, it is seen that the hyperparameters of the proposed model are well adjusted, and it gives successful results in both data sets. Classical machine learning algorithms have been able to give more successful results compared to the LSTM learning model due to the uneven distributions on the data sets and some classes with small data samples. However, when these datasets are expanded and larger samples are created, the LSTM architecture will be able to give successful results, as in the current NSL-KDD dataset.

**Table 4.** Comparison with latest state-of-the artwork.

| Author | Technique | Accuracy |
|---|---|---|
| Kazi Abu Taher et al. [33] | ANN | 95.00% |
| Mohammad N. I. et al. [34] | Decision tree, RF | 92.60% |
| Roberto M.-C. et [35] | Linear reg., random forest | 94.00% |
| Razan A. et al. [36] | RF, Bayesian Network, | 93.40% |

| | | |
|---|---|---|
| Proposed approach | Binary Class (UNSW-NB15, NSL-KDD) RF, MLP | 98.60%, 97.80%, |
| Proposed approach | Multi-Class (UNSW-NB15, NSL-KDD) LR, LSTM | 93.40%, 98.30% |

## 5. Conclusion

In this paper, machine learning algorithms for intrusion detection systems are run one by one. Applications are a guide for network attackers. The results obtained on two different data sets for binary and multi-class detection problems prove the success of machine learning algorithms in classification. In the UNSW-NB15 dataset, RF and LR algorithms gave the best results, respectively, and in the NSL-KDD dataset, the MLP and LSTM networks gave the best results. From this perspective, it can be said that almost all of the machine learning algorithms give good and close results. Deep learning and LSTM architectural structures were able to give successful results according to machine learning algorithms when large data sets and balanced class distributions were created. However, this study proves that classical machine learning algorithms are still a good alternative to deep learning models. Additionally, training and testing times were found to be close to each other. However, it is worth mentioning that attack classes are sometimes mislabeled.

Thanks to this and similar studies, it is possible to compare machine learning algorithms for network attacks. In addition, the option of detecting the superior and deficient aspects of algorithms and developing new hybrid systems accordingly is offered.

In future studies, the datasets can be improved, and the missing classes can be made more balanced. Additionally, new models can be designed by considering the superior features of machine learning algorithms. Considering that attack detection systems are constantly exposed to attacks, I believe that it would be beneficial to conduct such studies at short intervals and with updated data sets. I recommend that real-time web interfaces and intrusion detection systems be supported, where the best results algorithms will be used as drafts.

## Conflict of Interest Statement

The study is complied with research and publication Ethics

## Statement of Research and Publication Ethics

The study is complied with research and publication ethics.

## References

[1] S. Moualla, K. Khorzom, and A. Jafar, “Improving the Performance of Machine Learning-Based Network Intrusion Detection Systems on the UNSW-NB15 Dataset,” *Comput Intell Neurosci*, vol. 2021, pp. 5557577, 2021, doi: 10.1155/2021/5557577.

[2] S. Choudhary and N. Kesswani, “Analysis of KDD-Cup’99, NSL-KDD and UNSW-NB15 Datasets using Deep Learning in IoT,” *Procedia Comput Sci*, vol. 167, pp. 1561–1573, Jan. 2020, doi: 10.1016/J.PROCS.2020.03.367.

[3] B. M. Serinelli, A. Collen, and N. A. Nijdam, “Training Guidance with KDD Cup 1999 and NSL-KDD Data Sets of ANIDINR: Anomaly-Based Network Intrusion Detection System,” *Procedia Comput Sci*, vol. 175, pp. 560–565, Jan. 2020, doi: 10.1016/J.PROCS.2020.07.080.

[4] N. v. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/JAIR.953.

[5] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning 2006 63:1*, vol. 63, no. 1, pp. 3–42, Mar. 2006, doi: 10.1007/S10994-006-6226-1.

[6] A. Basati and M. M. Faghih, "PDAE: Efficient network intrusion detection in IoT using parallel deep auto-encoders," *Inf Sci (N Y)*, vol. 598, pp. 57–74, Jun. 2022, doi: 10.1016/J.INS.2022.03.065.

[7] A. E. Cil, K. Yildiz, and A. Buldu, "Detection of DDoS attacks with feed forward based deep neural network model," *Expert Syst Appl*, vol. 169, p. 114520, May 2021, doi: 10.1016/J.ESWA.2020.114520.

[8] G. C. Amaizu, C. I. Nwakanma, S. Bhardwaj, J. M. Lee, and D. S. Kim, "Composite and efficient DDoS attack detection framework for B5G networks," *Computer Networks*, vol. 188, p. 107871, Apr. 2021, doi: 10.1016/J.COMNET.2021.107871.

[9] G. A. MM, J. N. K. S, U. M. R, and M. R. TF, "An efficient SVM based DEHO classifier to detect DDoS attack in cloud computing environment," *Computer Networks*, vol. 215, p. 109138, Oct. 2022, doi: 10.1016/J.COMNET.2022.109138.

[10] E. Mushtaq, A. Zameer, M. Umer, and A. A. Abbasi, "A two-stage intrusion detection system with auto-encoder and LSTMs," *Appl Soft Comput*, vol. 121, p. 108768, May 2022, doi: 10.1016/J.ASOC.2022.108768.

[11] S. Choudhary and N. Kesswani, "Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 Datasets using Deep Learning in IoT," *Procedia Comput Sci*, vol. 167, pp. 1561–1573, 2020, doi: 10.1016/J.PROCS.2020.03.367.

[12] S. Moualla, K. Khorzom, and A. Jafar, "Improving the Performance of Machine Learning-Based Network Intrusion Detection Systems on the UNSW-NB15 Dataset," *Computational Intelligence and Neuroscience*, vol. 2021, 2021, doi: 10.1155/2021/5557577.

[13] L. Mohammadpour, T. C. Ling, C. S. Liew, and C. Y. Chong, "A convolutional neural network for network intrusion detection system," *Proceedings of the Asia-Pacific Advanced Network*, vol. 46, no. 0, pp. 50–55, 2018.

[14] A. Doğru, S. Buyrukoğlu, and M. Arı, "A hybrid super ensemble learning model for the early-stage prediction of diabetes risk," *Medical & Biological Engineering & Computing*, vol. 61, no. 3, pp. 785-797, 2023.

[15] S. Buyrukoğlu. "New hybrid data mining model for prediction of Salmonella presence in agricultural waters based on ensemble feature selection and machine learning algorithms," *Journal of Food Safety*, vol. 41, no. 4, 2021.

[16] S. Buyrukoğlu. "Promising cryptocurrency analysis using deep learning." *In 2021 5th International symposium on multidisciplinary studies and innovative technologies (ISMSIT)*," pp. 372-376, 2021.

[17] "The UNSW-NB15 Dataset | UNSW Research." https://research.unsw.edu.au/projects/unsw-nb15-dataset (accessed Sep. 08, 2022).

[18] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," *2015 Military Communications and Information Systems Conference, MilCIS 2015 - Proceedings*, Dec. 2015, doi: 10.1109/MILCIS.2015.7348942.

[19] S. Bagui, E. Kalaimannan, S. Bagui, D. Nandi, and A. Pinto, "Using machine learning techniques to identify rare cyber-attacks on the UNSW-NB15 dataset," *Security and Privacy*, vol. 2, no. 6, p. e91, Nov. 2019, doi: 10.1002/SPY2.91.

[20] P. TS and P. Shrinivasacharya, "Evaluating neural networks using Bi-Directional LSTM for network IDS (intrusion detection systems) in cyber security," *Global Transitions Proceedings*, vol. 2, no. 2, pp. 448–454, Nov. 2021, doi: 10.1016/J.GLTP.2021.08.017.

[21] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," *IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009*, Dec. 2009, doi: 10.1109/CISDA.2009.5356528.

[22] J. Mchugh, "Testing Intrusion detection systems," *ACM Transactions on Information and System Security (TISSEC)*, vol. 3, no. 4, pp. 262–294, Nov. 2000, doi: 10.1145/382912.382923.

[23] R. D. Ravipati and M. Abualkibash, "Intrusion Detection System Classification Using Different Machine Learning Algorithms on KDD-99 and NSL-KDD Datasets - A Review Paper," *SSRN Electronic Journal*, Jun. 2019, doi: 10.2139/SSRN.3428211.

[24] A. Karcioğlu and T. Aydin, "Sentiment Analysis of Turkish and English Twitter Feeds Using Word2Vec Model," *2019 27th Signal Processing and Communications Applications Conference (SIU)*, Sivas, Turkey, 2019, pp. 1-4, doi: 10.1109/SIU.2019.8806295.

[25] A. Moldagulova and R. B. Sulaiman, "Using KNN algorithm for classification of textual documents," *ICIT 2017 - 8th International Conference on Information Technology, Proceedings*, pp. 665–671, Oct. 2017, doi: 10.1109/ICITECH.2017.8079924.

[26] A. A. Akinyelu and A. O. Adewumi, "Classification of Phishing Email Using Random Forest Machine Learning Technique," *J. Appl. Math*, vol. 41, pp. 1-6, 2014, doi: 10.1155/2014/425731.

[27] H. Patel, P. Prajapati, and H. H. Patel, "Study and Analysis of Decision Tree Based Classification Algorithms Extreme Multi-label Classification Problem View project Significance of the Transition to Outcome Based Education: Explore the Future View project Study and Analysis of Decision Tree Based Classification Algorithms," *International Journal of Computer Sciences and Engineering Open Access Research Paper*, no. 6, 2018, doi: 10.26438/ijcse/v6i10.7478.

[28] W. H. Delashmit, "Recent Developments in Multilayer Perceptron Neural Networks".

[29] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/NECO.1997.9.8.1735.

[30] K. K. A. Ghany, H. M. Zawbaa, and H. M. Sabri, "COVID-19 prediction using LSTM algorithm: GCC case study," *Inform Med Unlocked*, vol. 23, Jan. 2021, doi: 10.1016/J.IMU.2021.100566.

[31] S. Tanışman, A.A. Karcıoğlu, U. Aybars and H. Bulut, "LSTM Sinir Ağı ve ARIMA Zaman Serisi Modelleri Kullanılarak Bitcoin Fiyatının Tahminlenmesi ve Yöntemlerin Karşılaştırılması," *Avrupa Bilim ve Teknoloji Dergisi*, vol. 32, pp. 514-520, 2021.

[32] S. Tanışman, A.A. Karcıoğlu, U. Aybars and H. Bulut, "Türkiye'de COVID-19 Bulaşısının ARIMA Modeli ve LSTM Ağı Kullanılarak Zaman Serisi Tahmini," *Avrupa Bilim ve Teknoloji Dergisi*, vol. 32, pp. 288-297, 2021.

[33] K.A., Taher, B.M.Y., Jisan, and M.M. Rahman, "Network intrusion detection using supervised machine learning technique with feature selection," *In 2019 International conference on robotics, electrical and signal processing techniques*, pp. 643-646, 2019.

[34] M. Injadat, A. Moubayed, A.B. Nassif, A. Shami, "Multi-stage optimized machine learning framework for network intrusion detection," *IEEE Trans. Netw. Serv. Manag*, 2020. Doi:10.1109/TNSM.2020.3014929

[35] R., Magán-Carrión, D., Urda, I., Díaz-Cano, and B., Dorronsoro, "Towards a reliable comparison and evaluation of network intrusion detection systems based on machine learning approaches," *Applied Sciences*, vol. 10, no. 5, p. 1775, 2020.

[36] R., Abdulhammed, H., Musafer, A., Alessa, M., Faezipour, and A., Abuzneid, "Features dimensionality reduction approaches for machine learning based network intrusion detection," *Electronics*, vol. 8, no. 3, p. 322, 2019.