Power System Technology
ISSN:1000-3673

Received: 06-02-2025 *Revised: 15-03-2025* *Accepted: 10-04-2025*

Designing a Network Intrusion Detection System based on Machine Learning for Software-Defined Networks

Shahabeddin Rahimi Harsini
Affiliation: University of Houston, Houston, Texas
Srahimih@cougarnet.UH.EDU
David Houshangi
Second writer email:dhoushan@cougarnet.uh.edu

Abstract
Today, the increasing development of computer networks and their wide application in human life have made it clear that these networks need to be secured. Therefore, various tools and equipment are used to ensure security, including the intrusion detection system. Given that most networks without a fixed infrastructure based on cloud computing face various security challenges, in recent years, different methods have used the distributed software-based network to deal with these challenges. This technology, while having many capabilities, faces vulnerabilities against some common threats and destructive factors, such as the distributed denial of service attack. A review of various studies shows that in order to eliminate vulnerabilities, the integration of defense solutions appropriate to the structure of the software-based network should be considered. Therefore, the aim of this research is to design a network intrusion detection system based on machine learning for software networks. This research was conducted with the XGBoost algorithm, which implements a decision tree with gradient boosting, which is designed for better speed and performance. In this method, we can have a Loss function that calculates the distance of the classifier from the final result. N trees are created and each one is assigned a coefficient, and the sum of these weights in each tree is the learning rate of that tree. Based on the research results, the proposed method of this research is to use XGBoost, which has shown very good performance compared to previous methods. Based on this method, it can be seen that the best method for intrusion detection and detection is to use the XGBoost algorithm, based on which an accuracy of over 92% can be achieved. With these interpretations, the proposed system can be practically implemented, given that it uses real and existing data sets, and it can be used in existing operational environments for error detection.

Key woeds: Network intrusion detection system, XGBoost algorithm, network security, machine learning.

Introduction

Today, the increasing development of computer networks and their widespread use in human life have made the need to secure these networks more evident than ever. Various tools and equipment are used to ensure security, including intrusion detection systems. Intrusion detection systems often use two methods of abuse detection and anomaly detection to detect intrusion. The use of modern architectures for intrusion detection systems has faced designers

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 Revised: 15-03-2025 Accepted: 10-04-2025

with difficulties in choosing the right type of architecture to create greater reliability in detecting attacks, and they have been forced to use complex designs to enhance the ability of these systems to detect attacks and remain immune to attacks against them. Also, in today's security world, unlike the past, database-based defense tools that define rules for detecting attacks do not have the necessary efficiency and have encountered problems in securing networks. Therefore, defense tools based on machine learning algorithms that have the ability to deal with the most complex types of attacks have been considered (1). Most cloud-based networks without a fixed infrastructure face various security challenges. In recent years, different methods have used distributed software-based networks to address these challenges. While this technology has many capabilities, it also faces vulnerabilities against some common threats and malicious factors, such as distributed denial of service attacks. A review of various studies shows that in order to address vulnerabilities, we need to integrate appropriate defense solutions with the distributed software-based network structure.

Cloud computing, as a convenient user interface, provides access to software resources for a large number of users through hardware located in a data center. Due to its extraordinary capabilities, cloud computing has attracted customers and huge investments and injects important features such as reliability, cost reduction, multi-tenancy, security, etc. into various networks. One of the most important security issues now is the challenges of networks without fixed infrastructure and centralized management based on it (2). In some studies, the use of software-defined network features has been proposed to improve security in cloud-based networks. This technology brings capabilities such as planning, centralized control, and traffic analysis to the network and can be used to improve security. Virtualization also adds intelligence and new perspectives to the network (2). Due to the growth of heterogeneous wireless technology, software-less networks based on infrastructure also provide flexibility, dynamism, scalability and opportunities to solve security problems. Software-defined networks, while introducing planning, centralized control, traffic analysis and security improvements, create virtualization and intelligence in the network. Given that the use of a single-level approach with centralized control is incompatible with decentralization, sabotage and delay in environments without infrastructure, its application faces challenges (3). The basis of the work of software-defined networks is the separation of network intelligence (control) from the shell (information transmission). Such networks became popular after the release of Java by Sun Microsystems in 1995. In the above networks, a centralized controller manages the entire network, and programming allows for network dynamics, personalization and the definition of new applications based on the needs of each organization (4).

Intrusion detection systems are classified into two groups based on their location in the network system and scope of activity: host-based intrusion detection systems (HIDS) and network-based intrusion detection systems (NIDS). Usually, to achieve maximum efficiency and security, these intrusion detection systems are also used in combination, which are known as distributed intrusion detection systems (DIDS) (9). In fact, intrusion detection systems are designed to identify user activities in both normal and automatic ways by comparing network connection transactions and based on known intrusion patterns designed by specialists and experts. Traditional methods cannot efficiently discover unknown intrusion patterns. Because human forces encounter networks of computer systems that are fast and

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 Revised: 15-03-2025 Accepted: 10-04-2025

complex during intrusion detection analysis. Therefore, intelligent decision-making techniques and technologies based on data mining are used in this regard to be able to identify the pattern. or identify effective and efficient patterns in detecting intrusions. A secure computer system is a system that behaves in a way that is expected of it.
A more precise definition of a secure computer system is as follows. It is a secure system that can provide confidentiality, integrity, and availability to its users. Confidentiality means that information is only available to those who have been authorized to access that data by the system administrator, and in this regard, integrity means that the information remains unchanged due to accidents or sabotage. Availability also means that the system is active when needed and provides users with information without any problems. In fact, intrusion includes a set of illegal actions that compromise the integrity, confidentiality, or access to a resource (10).

Network-based intrusion detection systems monitor and analyze network-wide traffic in order to identify threats. These systems detect malicious activities such as denial of service (DOS) attacks, so-called port attacks, etc. across the entire network. Network-based intrusion detection systems are hardware or software that are placed on the network to monitor traffic passing through the entire network and computers in the network are placed in specific locations on the network, analyze the network traffic, and when an attack or unusual behavior is detected, an alert message is sent to the network administrator. Network-based intrusion detection systems are independent of the operating system because they operate at the network layer level. These intrusion detection systems examine network traffic for each packet passing through the network in real time or near real time to identify intrusion patterns (11).
One of the challenges that network-based intrusion detection systems face is that these systems examine bytes, packets, and network flow to detect intrusions, and in high-speed networks, it is impossible to monitor all packets. These systems are able to collect data at a speed of They are not high and have problems in networks with speeds of 111Mbps and lose many packets in transit. Also, attacks with encrypted communication are increasing and the content of encrypted packets is inaccessible to the intrusion detection system. To solve this problem, analyzing network flows instead of analyzing each packet is the best option for monitoring and analysis (12). The main objective of this research is to design a network intrusion detection system based on machine learning for software networks. The main assumption in this research is that it is possible to define a network intrusion detection system based on machine learning for software networks.

Theoretical Concepts and Research Background
Intrusion Detection System To describe network security and the attacks that are carried out on it, it is first necessary to know the necessary definitions of security. For this purpose, we will explain several definitions of security. The first definition is the presence or feeling of security. The second definition is the safety of a state or organization against criminal and espionage activities. On the other hand, security is often considered as the art of sharing secrets. Cryptographers often use this definition for security. Today, we define security in the

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 Revised: 15-03-2025 Accepted: 10-04-2025

digital age as an effort to protect digital assets and protect the system from any activity that is done or seen unintentionally. This unwanted behavior is abuse and violation of the property of individuals (1). In a system like the Internet, interconnected, standardized, open and managed by different people, under different legal systems, attacks are very common [1]. Classification of types of attacks • Network device level attack A network device level attack exploits weaknesses in the hardware used in the network or bugs and weaknesses in the software. For example, some routers have a buffer overflow problem that can be used by an attacker to disable the router. • Operating system level attack The second category of operating system level attacks is the one that exploits some aspects of the protocol implementation as well as the operating system of the target system. • Application level attack Application level attacks exploit holes in application programs. For example, this type of attack can exploit flaws in the data structure of algorithms used in some common programs. For example, by giving special inputs, the algorithm can be forced to run in an unexpected state and cause it to malfunction. This causes the system to allocate most of the CPU to perform the necessary calculations and therefore not handle a large portion of the incoming traffic. • Data flood level attack The attacker achieves his goal of taking up system resources and bandwidth by sending a flood of packets with a fake source address and at a high rate. This keeps the system busy processing these fake packets and It prevents the inspection of useful and original data. • Protocol-level feature attack This attack exploits the insecure design of standard protocols. For example, the SYN flood attack exploits the three-way handshake process in the TCP protocol and the fact that the source IP address is not checked for validity in the routing process. Main attacks • Remote attacks In a remote attack, the attacker sends a flood of unsolicited and malicious packets to a serving node over multiple hops. When a routing node receives these packets, it checks their routing table and finds that the entry on this route matches the destination address; therefore, it forwards them. If the node follows the route backward based on the source address, it may reach the claiming source instead of the flooding source, or it may find that the claiming node is invalid [5]. • Local Attacks In a local attack, attackers send flood traffic to their neighboring nodes to affect traffic through neighboring nodes. One advantage of local attacks is that the flooding nodes do not need to send traffic through multiple hops, so the flooding nodes do not rely on other routing nodes. In addition, attackers may attack a large area instead of a single node in order to maximize the impact on the entire network.

The figure below shows remote attacks and local attacks from DDoS attacks, where the gray oval area is a network where nodes a_1, a_2, a_3 are attackers and nodes n_1, n_2, n_3 are authorized nodes. The dotted lines indicate attack traffic through multiple nodes and the solid lines indicate attack traffic to neighboring nodes. The hatched areas are the areas of concentration due to the attack.

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 Revised: 15-03-2025 Accepted: 10-04-2025

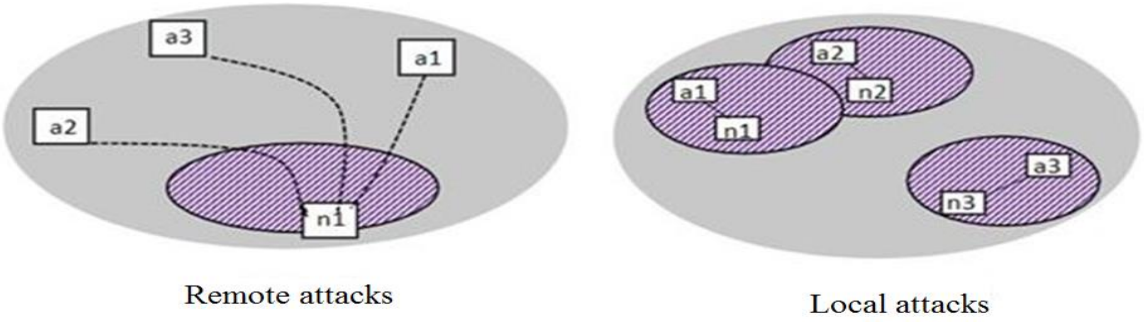

Figure . DDoS attacks based on attack topology(8)

Flood Attacks In this type of attack, the zombie(s) sends a large volume of traffic to the victim system in order to occupy its bandwidth. This stream of packets sent to the victim system causes it to slow down or crash, or saturate the network bandwidth. Several types of these attacks are, ICMP flood, UDP flood, and HTTP flood[6-8] • Amplification Attacks The attacker or agents use the P-address broadcast feature available in most routers to amplify the attack and send messages to a P-address. In this method, the router sends network packets to all IP addresses within the range of the broadcast address, which causes all subnets with the broadcast address to send a response to the victim system. As a result, the traffic generated reduces the bandwidth of the victim system. In this type of DDoS attack, the attacker can send the broadcast messages directly or by using agents to increase the attack traffic. If the attacker sends the public messages directly, this allows the attacker to use the systems on the network with the public IP address as zombies without having to hack into them or install any agent software. Two types of such attacks are smurf and fraggle.[6-8] Machine learning To solve a problem on a computer, we need an algorithm. An algorithm is a sequence of instructions that must be executed to transform input into output. For some tasks, for example, sorting a set of numbers, there are one or more algorithms. While for some other tasks, for example, distinguishing spam from other mail, there is no algorithm but sample data is available. There is a certain pattern in the data that the computer (machine) can identify a good and useful approximation for. In other words, the computer (machine) can automatically derive an algorithm for performing such tasks using sample data. For example, a computer can “learn” to distinguish spam from other emails by analyzing thousands of emails that are known to be spam or not. This is the concept of machine learning[11]. Machine learning methods can be divided into several general categories: 1. Supervised learning 2. Unsupervised learning 3. Semi-supervised learning 4. Reinforcement learning. Software-defined networking Software-defined networking technology is a new approach and a major development in the field of communication technology, aiming to optimize the use of resources, improve performance and productivity, and reduce costs in computer networks, as a centralized, cost-effective, adaptable, and manageable architecture suitable for dynamic services with high bandwidth in the coming years. Software-defined networks are trying to increase the intelligence of networks and provide capabilities such as planning, scalability,

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 Revised: 15-03-2025 Accepted: 10-04-2025

flexibility, automation, intelligence and software development of the network by organizations by transferring the data control part from hardware switches and routers to virtual software layers of the network and using a centralized software controller. This idea was first proposed in 2005 and gained momentum since 2010 and entered a new phase in 2011 with the formation of the ONF Foundation and the membership of more than eighty large companies in the network industry and the development of the OpenFlow standard. The first functional products of this network have been on the market since 2013 and it is predicted that these types of networks will gradually replace traditional networks based on Ethernet and TCP/IP [40]. The figure below shows the single-controller and multi-controller architectures in software-defined networks.

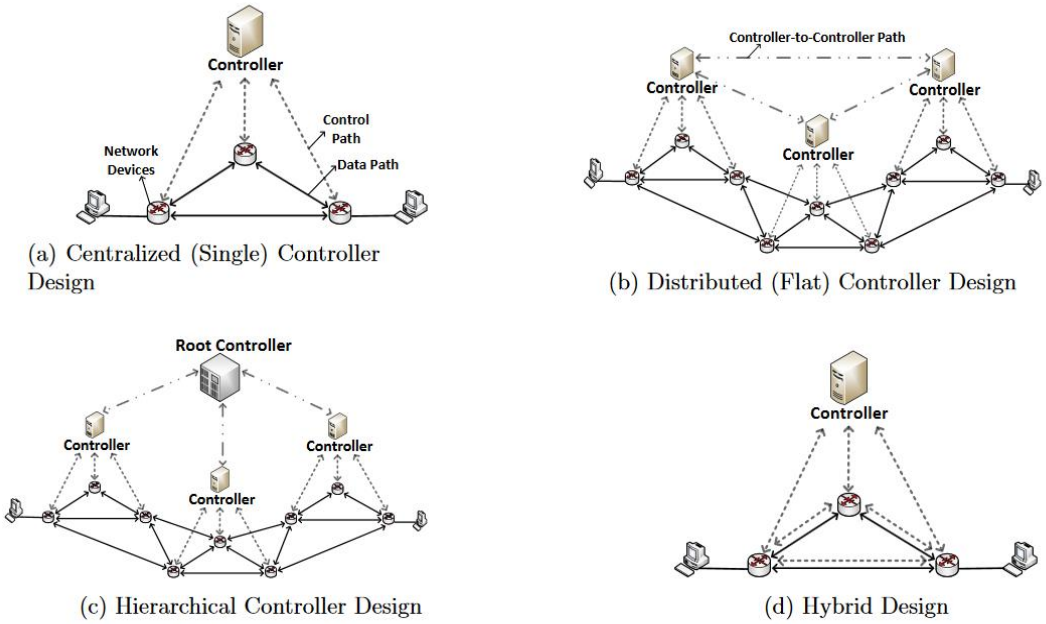

Fig. Single-controller and multi-controller architecture in software-defined networks[[46

One of the important advantages of software-defined networking is that it has attracted the attention of many researchers in this field. The benefits of intelligent dynamic load balancing in network traffic engineering include the following: [[58

•Optimization of operational throughput and increased throughput and improved efficiency of network resources

•Distribution and division of traffic load on all communication links in symmetric or asymmetric ways

•Control of heavy and light data flows and traffic management and prevention of traffic congestion, congestion and deadlocks

•Increase communication capacities by link aggregation and the possibility of more simultaneous connections and increased reliability and stability

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 Revised: 15-03-2025 Accepted: 10-04-2025

•Prevention of overload and management and guidance of excess load overflow and its fair distribution

•Control and improvement of failure and fault tolerance threshold and network resilience and increased stability

•Intelligent detection of service outage and service distribution and migration to establish communication without loss and interruption in the network

•Intelligent detection of the addition of a service or channel and its management without loss and interruption in the network

•More stable communication in the network by providing backup and alternative paths in the event of a problem in the path or service provider

•Reduction of end-to-end delay time and reduction of the total work completion time

•Cost reduction In purchasing resources and developing hardware and operations due to optimal use of resources and economic savings

•Reducing the cost of ownership of lines for users due to optimal use of shared resources

•Increasing the ability and ease of accessibility by connecting to multiple servers and preventing network drops [[49

•Improving the quality of service and increasing user satisfaction [[46

•Preventing bottlenecks and points of failure in the network

•Increasing security and reliability in service provision [[57

•Improving scalability and scalability

•Fairness, justice, and equality (not equality) in the use of channels and resources for all service providers and service recipients

Research Background

Chopani Sefidi (1400) in an article titled "Software-Based Network Intrusion Detection to Increase Accuracy and Reduce Attacks" In this thesis, an attempt has been made to increase accuracy and reduce attacks by using a combination of support vector machine and k-nearest neighbor algorithms. Using machine learning algorithms to classify connections into legitimate and illegitimate is one of these solutions. (14) Namjooi Rad et al. (1399) in an article titled "Network Intrusion Detection Using Data Mining and Using Machine Learning Using Support Vector Machine Method" In this article, they examine the support vector machine algorithm in feature selection and the effect of using machine learning algorithms on the accuracy and rate of intrusion detection in the system. The results show that using this algorithm leads to an increase in accuracy and correct detection of alerts compared to previous methods (15). Alipour et al. (2019) in an article titled "A Review of Software-Based Network-Based Intrusion Detection Defense Solutions" presented a general classification of the types of defense solutions against the above attacks. In the following, while classifying intrusion detection solutions into two categories: threshold and non-threshold, they examined some practical examples of the above solutions and concluded that the threshold nature of the intrusion detection method exacerbates the level of vulnerability and non-threshold defense solutions with a flat, distributed software-based network architecture should be used (16). Akhlaqpour (2018) in an article titled "Presenting a Machine Learning-Based Intrusion Detection System to Reduce Network Attacks Based on Clustering" presented a machine

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 Revised: 15-03-2025 Accepted: 10-04-2025

learning-based intrusion detection system that is capable of detecting common network attacks including denial of service, botnets, intrusion, and network scanning. With the help of the proposed intrusion detection system, it has been shown to what extent attacks and training (and more complex types of them) affect machine learning-based detection systems and how they can be detected (17). Torabi et al. (2019) in an article titled "Design and Implementation of a Trust-Based Framework for Mobile Traffic Detection in the Network" In this article, a framework for mobile traffic detection with a trust-based approach is presented. The proposed framework consists of four detection subsystems in different TCP/IP layers and two control subsystems. First, with the help of the environment detection subsystem, the accuracy of the information related to each layer is checked, and if the information of each layer is valid, the traffic is sent to the detection subsystems. After the detection in each of the subsystems, based on the factors obtained in each subsystem, the final detection of each agent is performed by the detection subsystem. In this paper, in order to evaluate the proposed framework and examine the existing challenges, a real dataset of wireless network traffic (Ferdowsi University of Mashhad) has been prepared and used. The results show that the proposed system is able to perform the detection process with a maximum accuracy of 0.97 and a minimum error of 0.09 by integrating the opinions of the detection subsystems based on the level of trust and confidence in each of them (18).

In 2017, Yen et al. studied how to model a deep learning-based intrusion detection system and proposed a deep learning method for intrusion detection using recurrent neural networks (RNN-IDS). In addition, they studied the performance of the model in binary and multi-class classes and the number of neurons and the effects of different learning on the performance of the proposed model. They compared their method with artificial neural network, random forest, support vector machine and other machine learning methods proposed by previous researchers. Their experimental results show that the proposed model (RNN-IDS) is a suitable and high-accuracy classification model for modeling, and its performance is better than that of traditional machine learning classification methods. They believed that the proposed model (RNN-IDS) improves the accuracy of intrusion detection and provides a new research method for intrusion detection[25]. In 2018, Diro et al. conducted a new deep learning approach to enhance cybersecurity for detecting attacks in the Internet of Things. The performance of the deep learning model is evaluated in comparison to the traditional machine learning method, and the distributed attack detection against the centralized detection system. These experiments have shown that the distributed attack detection system using the deep learning model is superior to the centralized detection systems. It has also been shown that the deep learning model is more effective in detecting attacks than the traditional machine learning method[26]. Anurag et al. (2022) in a paper titled “Network Intrusion Detection in Software Defined Networks with a Self-Organized Constraint-Based Intelligent Learning Framework” Therefore, the aim is to investigate and present effective and efficient intrusion detection techniques that solve the security challenges posed in the field of software-based IoT systems, a wide range of errors such as DOS, DDOS, U2R, etc. Abdulsalam et al. (2022) in a paper titled “ML-IDSDN: A Machine Learning-Based Intrusion Detection System for Software-Defined Networks” is used to train a number of supervised binary classification machine learning algorithms such as k-nearest neighbor, AdaBoost,

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 Revised: 15-03-2025 Accepted: 10-04-2025

Decision Tree (DT), Random Forest, Naive, Multilayer Perceptron, Support Vector Machine and XGBoost. And the results of the DT algorithm have achieved high scores for adapting a real-time application by obtaining an F1 score in the attack class of 0.9995, an F1 score in the normal class of 0.9983 and a throughput score of 6737147.275 samples per second with a total number of three features. Oqbah et al. (2020) in a paper titled "Machine Learning-Based Intrusion Detection System for Software-Defined Networks". In this research, a new anomaly-based IDS is proposed that takes advantage of this software capability to provide statistical features about each flow passing through the network, and transfers these features to a voting system. In this regard, several machine learning algorithms are used to predict user behavior towards possible intrusions, and the results show an increase in accuracy and a decrease in false positive rate in the training voting software using the NSL-KDD and KDDCup99 datasets. Abu Bakr et al. (2017) in their paper titled "Machine Learning-Based Intrusion Detection System for Software Defined Networks" focus is shifted to a single point of failure where the central controller is the main target. The results show a positive improvement for detecting almost all possible attacks in the SDN environment with our neural network pattern recognition for machine learning using our trained model with an accuracy of over 97%.

Based on a review of the history of research conducted on the issue of a network intrusion detection system based on machine learning for software-defined networks, it can be said that research on the design of a network intrusion detection system based on machine learning for software-defined networks is necessary.

Research Methodology

The XGBoost algorithm is an implementation of a gradient boosting decision tree that is designed for better speed and performance. In this method, we can have a Loss function that calculates the distance between the classifier and the final result. N trees are created and each is assigned a coefficient, and the sum of these weights in each tree is the learning rate of that tree.

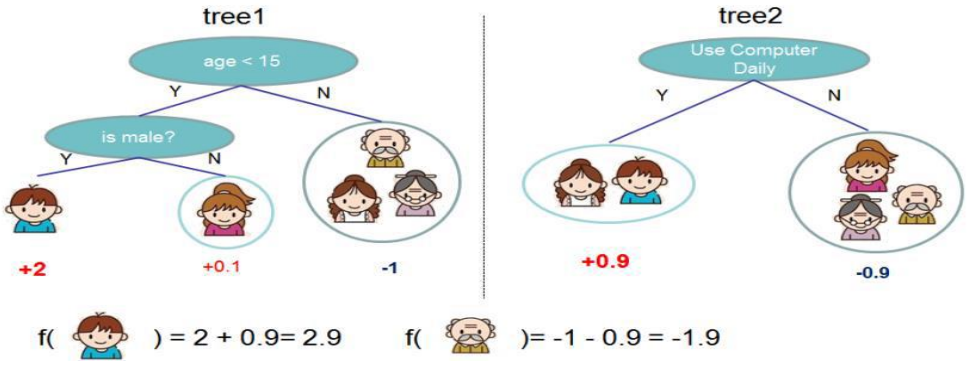

Figure . XGBoost method

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 *Revised: 15-03-2025* *Accepted: 10-04-2025*

Here, the decreasing gradient is the decreasing gradient that acts inversely to the derivative and computes the minimum.

XGBoost Features:

Three main forms of gradient boosting attacks are supported:

•Gradient boosting

•Random gradient boosting

•Regular gradient boosting

In terms of computation, it provides the following:

•Partitioned tree construction using all CPU cores during training

•Distributed computation for training very large models using a cluster of resources

•Out-of-core computation for very large data sets that do not fit in memory

•Data structure and algorithm optimization for best use of hardware

This algorithm is designed to make use of available resources for training the model. A deliberate sparse implementation with automatic handling of missing data, a block structure to support parallelization of tree construction, and continuous training are other features of this algorithm.

Regarding the feature selection process, a large number of classification techniques have been proposed by the scientific community over the years, and the selection of the most appropriate classifier for a given task is often improved by previous experience in different domains, as well as by trial and error methods. However, recently, some researchers evaluated the performance of about 180 classifiers obtained from different families using different datasets and concluded that random forests and support vector machines are the two classification mechanisms that have the highest probability of performing well. On the other hand, many winners in recent Kaggle competitions use the XGBoost technique, which is a parallel component of gradient boosting of tree classifiers, and it can be said that it performs better than random forests in most cases. The XGBoost technique is a library that is known as a parallel, fast and efficient algorithm whose parameters are fully adjustable. The high performance and reliability of XGBoost were the reasons for using this library for the experiment at hand.

```
[ ]  XGB = xgboost.XGBClassifier(n_estimators=100, learning_rate=0.017, gamma=0,
```

```
[ ]
     pred = cross_val_predict(XGB, X, Y,  cv=5)
     acc = accuracy_score(Y, pred)
     cm = confusion_matrix(Y, pred)
     print(acc)
     print(cm)
```

Figure . Schematic of XGBoost code

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 *Revised: 15-03-2025* *Accepted: 10-04-2025*

In fact, the XGBoost algorithm method and how it works were explained in this chapter. How it is implemented for intrusion detection systems in software-based networks was also discussed.

Results and Discussion

NSL-KDD is a dataset proposed to solve some of the inherent problems of the KDD99 dataset. However, this new version of the KDD dataset still suffers from some of the problems discussed by McHugh and may not be fully representative of existing real networks, due to the lack of a public dataset for network-based IDSs, which can still be used as an effective benchmark dataset to help researchers compare different intrusion detection methods. In addition, the number of records in NSL-KDD and the test sets is reasonable. This advantage makes it cost-effective to run tests on the full set without having to randomly select a small portion. As a result, the evaluation results of different research work will be consistent and comparable. KDDTrain+.ARFF: Complete NSL-KDD set with binary labels in ARFF format KDDTrain+.TXT: Complete NSL-KDD set including attack type and difficulty level labels in CSV format KDDTrain+_20Percent.ARFF: A 20% subset of the KDDTrain+.arff file KDDTrain+_20Percent.TXT: A 20% subset of the KDDTrain+.txt file KDDTest+.ARFF: Complete NSL-KDD test set with binary labels in ARFF format KDDTest+.TXT: Complete NSL-KDD test set including attack type and difficulty level labels in CSV format KDDTest-21.ARFF: A subset of the KDDTest+.arff file that does not include records with difficulty level 21 out of 21. KDDTest-21.TXT: A subset of the KDDTest+.txt file that does not include records with difficulty level 21 out of 21. The NSL-KDD dataset has the following advantages over the original KDD dataset: • It does not contain redundant records in the dataset, so the classifiers are not biased towards repeated records. • There are no repeated records in the proposed test sets. Therefore, the performance of learners is not biased towards methods that have better recognition rates on repeated records. • The number of records selected from each difficulty level group is inversely proportional to the percentage of records in the original KDD dataset. As a result, the classification rates of distinct machine learning methods vary over a wider range, which makes it more efficient to accurately evaluate different learning techniques. • The number of records in the dataset and test sets is reasonable, which makes it cost-effective to run experiments on the entire set without having to randomly select a small portion. As a result, the evaluation results of different research works will be consistent and comparable. One of the most important shortcomings of the KDD dataset is the large number of redundant records, which causes the learning algorithms to be biased towards frequent records and thus prevent learning of non-repeated records, which is usually detrimental for networks such as U2R. R2L attacks In addition, the presence of these frequent records in the test set biases the evaluation results by methods that have a better detection rate on frequent records. In addition, we analyzed the difficulty level of the records in the KDD dataset. Surprisingly, about 98% of the records in the data set and 86% of the records in the test set were correctly classified by all 21 learners. In order to conduct our experiments, we randomly created three smaller subsets of the KDD dataset, each containing fifty thousand records of information.

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 *Revised: 15-03-2025* *Accepted: 10-04-2025*

Each of the learners was trained on the created datasets. We then used 21 trained machines (7 learners, each trained 3 times) to label the records of the entire KDD train and test sets, providing us with 21 default labels for each record. In addition, we annotated each record of the dataset with a #successfulPrediction value, which was initially set to zero. Now, since the KDD dataset provides the correct label for each record, we compared the predicted label of each record given by a particular learner with the actual label. Where if a match was found, we incremented the #successfulprediction by one. Through this process, we calculated the number of learners who were able to correctly label that record. The highest value for #successfulprediction is 21, which indicates the fact that all learners were able to correctly predict the label of that record.

The dataset data in the training and testing sections are shown in the figures below.

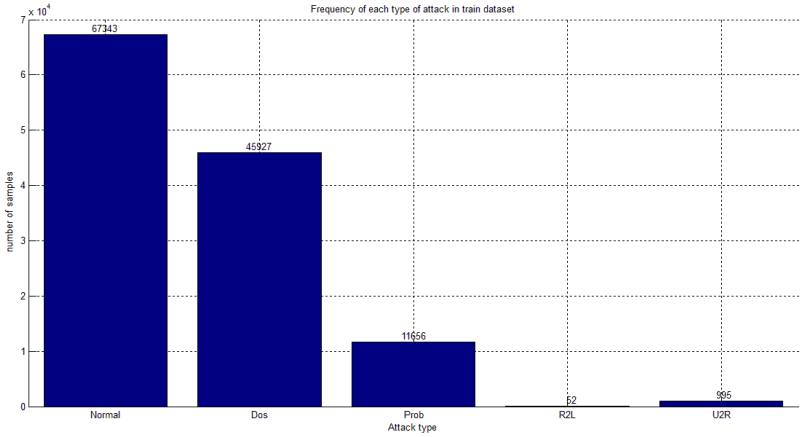

Fig. Training data

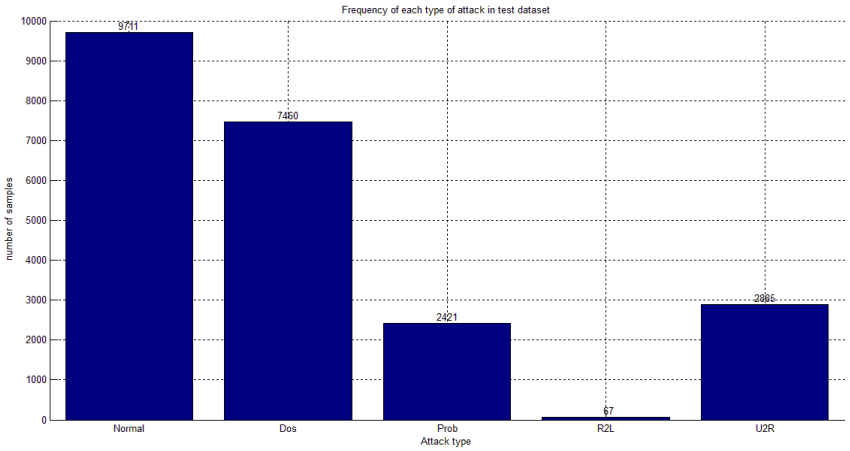

Fig. Test data

The following figures show the results of applying the methods to the NSL-KDD dataset.

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 *Revised: 15-03-2025* *Accepted: 10-04-2025*

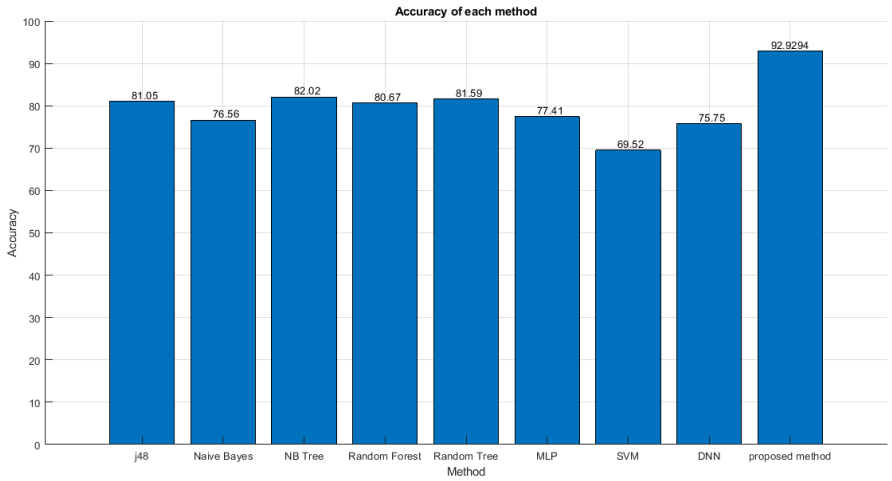

Fig. Accuracy of different methods compared to the proposed method

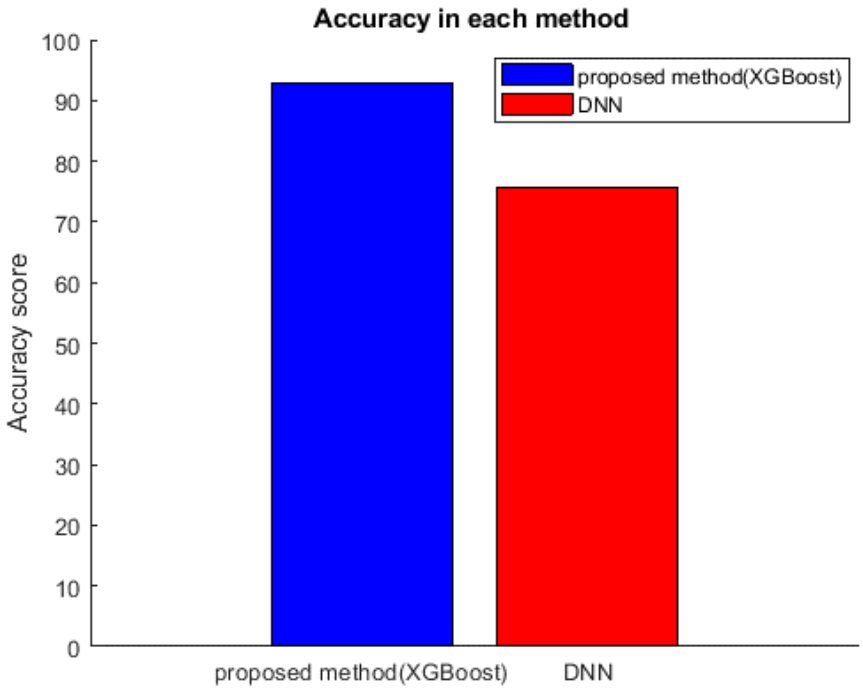

Fig. Accuracy of the proposed method compared to deep neural network

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 *Revised: 15-03-2025* *Accepted: 10-04-2025*

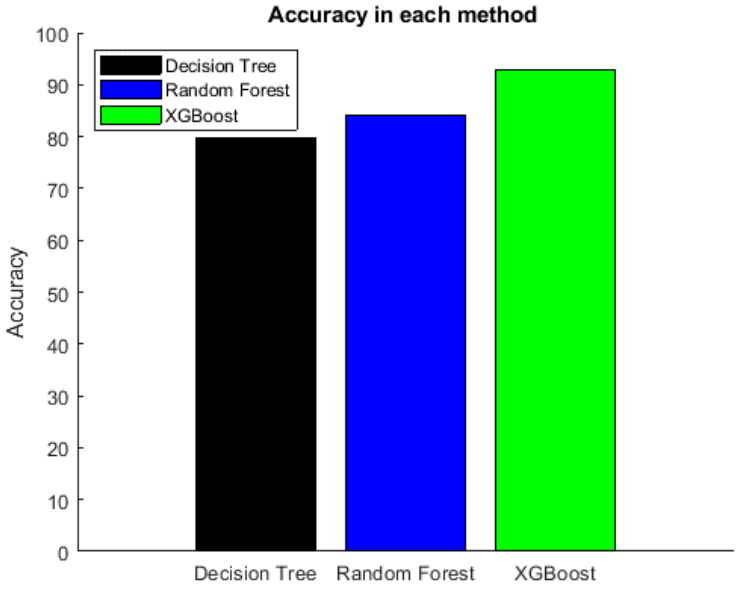

Fig. Accuracy of the proposed method compared to decision tree and random forest

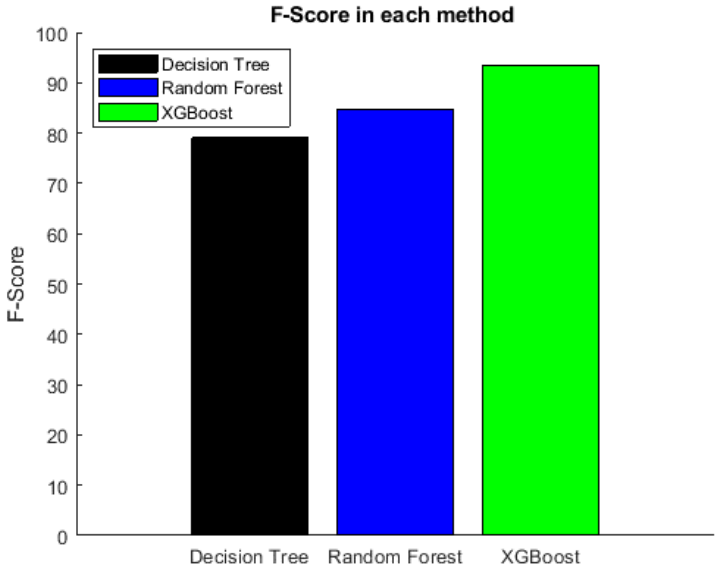

Figure . f1 score compared to decision tree and random forest.

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 *Revised: 15-03-2025* *Accepted: 10-04-2025*

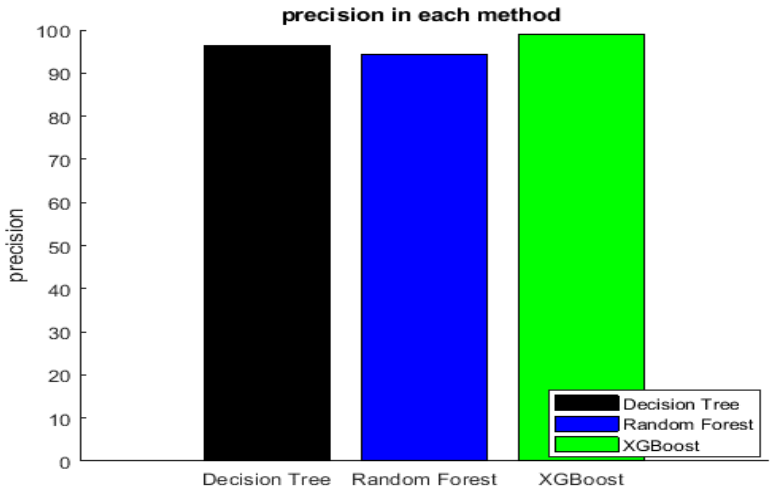

Figure . Accuracy of the proposed method compared to decision tree and random forest.

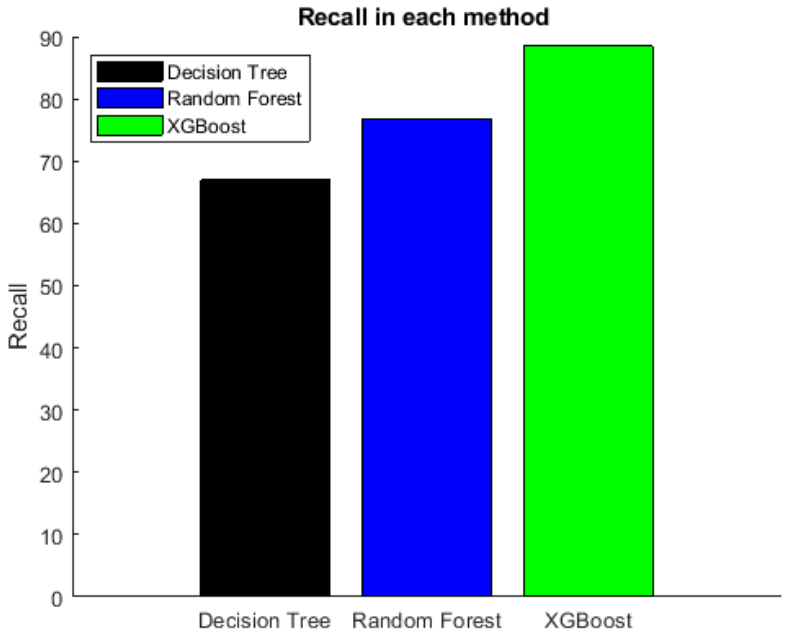

Figure . Invocation of the proposed method compared to decision tree and random forest.

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 *Revised: 15-03-2025* *Accepted: 10-04-2025*

The system performance characteristic curve using decision tree, random forest, and the proposed method is shown in the figures below

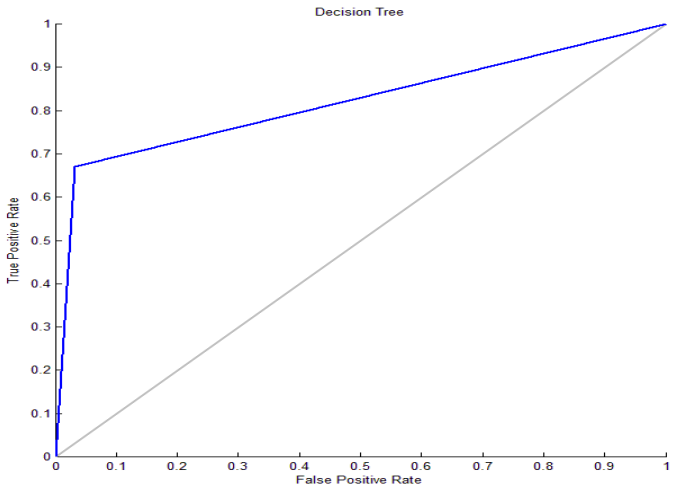

Figure . System performance characteristic curve in a decision tree

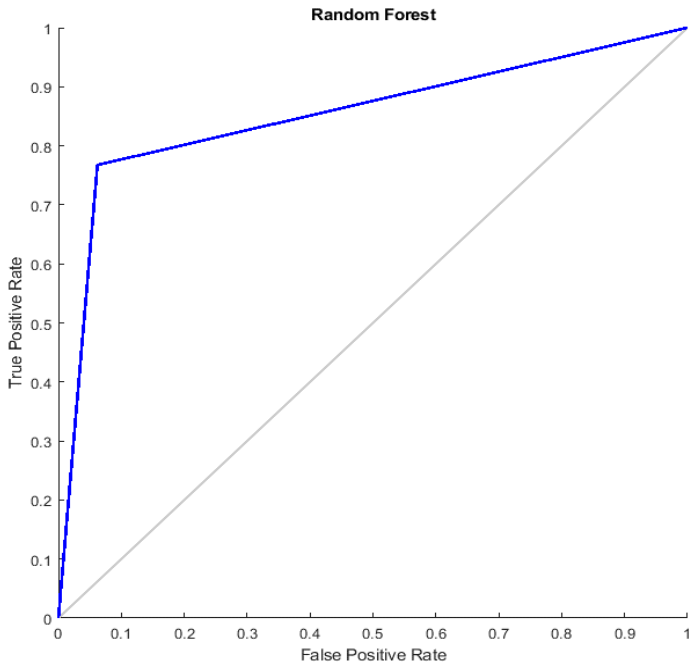

Figure . System characteristic performance curve in random forest.

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 *Revised: 15-03-2025* *Accepted: 10-04-2025*

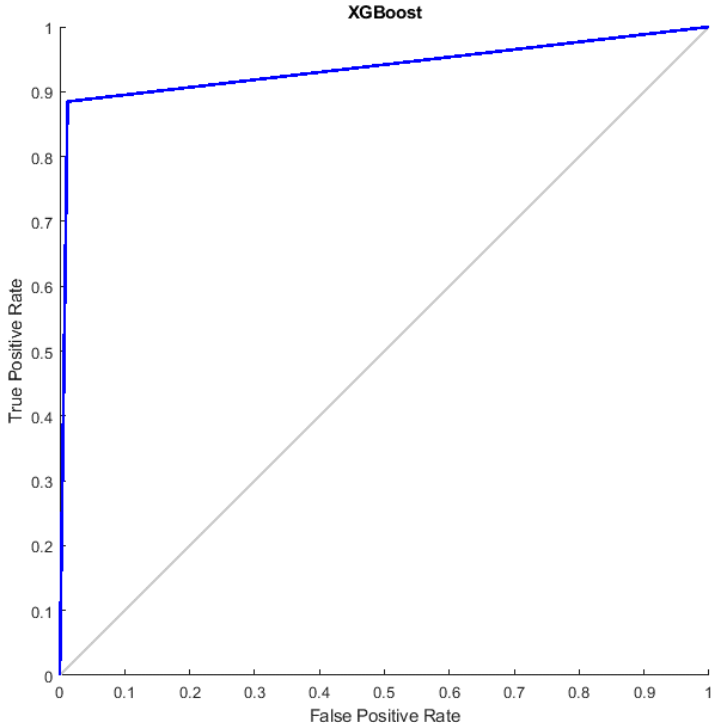

Fig. System performance characteristic curve in the proposed method based on XGBoost

Based on the research results, it can be said that the proposed method is more efficient than other existing methods.

Conclusion

The use of software-defined networks is increasing day by day due to its appropriate efficiency in various places such as homes, cities, etc., but these networks are challenged in terms of security and attacks on these networks may occur at any time. For this purpose, this research sought to improve the security challenge in software-defined networks. The proposed method of this research is the use of XGBoost, which showed very good performance compared to previous methods. According to the studies conducted and the proposed method, it can be seen that the best method in detecting and detecting intrusions is the use of the XGBoost algorithm, which achieved an accuracy of over 92 percent. With these interpretations, the proposed system can be practically implemented, given that we have used real and existing data sets, and it can be used in existing operational environments for error detection.

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 *Revised: 15-03-2025* *Accepted: 10-04-2025*

Suggestions

Also, by optimizing the XGboost algorithm, the detection accuracy of the proposed method can be increased to increase its reliability, and the efficiency of the system can be increased to perform this task at a more appropriate time and prevent infiltration. As another suggestion, by increasing the extracted features and also detecting optimal features, the system's performance in infiltration detection and its accuracy can be increased.

References

[1]H. Chaouchi and M. Laurent-Maknavicius, "Wireless and Mobile Networks Security", ISTE LTD, 2009.
[2].G. Sharma, S. Bala, and A. K. Verma, "Security Frameworks for Wireless Sensor Networks-Review", Proceedings of the International Conference on Communication, Computing & Security (ICCCS), Bhubaneswar, Vol. 6, pp. 978-987, 2012.
[3] D. G. Padmavathi and M. Shanmugapriya, "A survey of attacks, security mechanisms and challenges in wireless sensor networks", arXiv.org, 2009.
[4]M. Stamp, "Information Security: Principles and Practice", John Wiley & Sons, New York, 2011.
[5]. Y. C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks", Journal of Wireless networks, Vol. 11, pp. 21-38, 2005.
[6].F. Lau, S. H. Rubin, M. H. Smith, and L. Trajkovic, "Distributed denial of service attacks", Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Nashville, TN, Vol. 3, pp. 2275-2280, 2000.
[7]. S. Chithra and E. G. D. P. Raj, "Overview of DDoS algorithms: a survey", International Journal of Computer Science and Mobile Computing (IJCSMC), pp. 207-213, 2013.
[8]. B. Q. M. AL-Musawi, "Mitigating DoS/DDoS attacks using iptables", International Journal of Engineering & Technology, Vol. 12, pp. 101-111, 2022.
[9].S. M. Specht and R. B. Lee, "Distributed Denial of Service: Taxonomies of Attacks, Tools, and Countermeasures", Proceeding of the 17th International Conference on Parallel and Distributed Computing Systems (ISCA PDCS), San Francisco, California, USA, pp. 543-550, 2004.
[10].S. M. Bellovin, "Security problems in the TCP/IP protocol suite", Proceedings of the Association for Computing Machinery the Special Interest Group on Data Communication (ACM SIGCOMM) Computer Communication Review, Salvador, Brazil, Vol. 19, pp. 32-48, 1989.
[11] E. Alpaydin, Introduction to machine learning, 2010.
[12] P. a .Cunningham, M. Cord, and S. J. Delany, Supervised Learning, 2008.
[13] X. Zhu, Semi-Supervised Learning Literature Survey, 2008.
[14] A. Zell ،“Simulation neuronaler netze simulation of neural networks،” ۰۳ ۱۹۹۴.
[15] A. R. Mehryar Mohri and A. Talwalkar ،“Foundations of machine learning،” ۰۵ .۲۰۱۲.
[16] W. Zhang ،“Shift-invariant pattern recognition neural network and its optical architecture،” ۰۲ .۱۹۸۸
[17] K. Fukushima ،“Neocognitron”. scholarpedia،” .۲۰۰۷

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 *Revised: 15-03-2025* *Accepted: 10-04-2025*

[18] T. N. Hubel ،D. H.؛Wiesel ،"Receptive fields and functional architecture of monkey striate cortex" ،the journal of physiology،" .١٩۶٨
[19] Tarabalka, Yuliya; Chanussot, Jocelyn; Benediktsson, Jon Atli. Segmentation and classification of hyperspectral images using watershed transformation. Pattern Recognition, 2010, 43.7: 2367-2379.
[20] H.Habibi ،Aghdam ،"Guide to convolutional neural networks : a practical application to traffic-sign detection and classification،" .٢٠١٧
[21] "Convolutional neural networks for visual recognition،" .٢٠١٨
[22] K. Ch'ng ،J. Carrasquilla ،R. G.Melko ،and E. Khatami ،"Machine Learning Phases of Strongly Correlated Fermions،" Physical Review X ،vol. ٧، p. ٠٣١٠٣٨، Jul .٢٠١٧
[23] A. Decelle ،V. Martin-Mayor ،and B. Seoane ،"Learning a Gauge Symmetry with Neural Networks،" arXiv e-prints ،p. arXiv:١٩٠۴. ، ٠٧۶٣٧Apr.٢٠١٩
[24]Zachary Chase Lipton. "The Mythos of Model Interpretability". In: CoRR abs/1606.03490 (2016). arXiv: 1606.03490. URL: http://arxiv.org/abs/1606.03490.
[25] Yin, C., et al., A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access, 2017. 5: p. 21954-21961.
[26] Diro, A.A. and N. Chilamkurti, Distributed attack detection scheme using deep learning approach for Internet of Things. Future Generation Computer Systems, 2018. 82: p. 761-768.
[27]Choupani Sefidi, Mina; Salehi, Pouyan; Shayan, Farshid. (1400). Software-based network intrusion detection to increase accuracy and reduce attacks, Sixth National Conference on New Approaches in Education and Research.
[28]Namjooi Rad; Amir Abbas; Dadgarpour, Mehdi. (2010). Network intrusion detection using data mining and using machine learning using support vector machine method. Karafan Quarterly, 17(4):13-34.
[29]Alipour, Masoud; Shokrollahi, Saeed. (2010). Review of defense solutions for software-based network intrusion detection, and the scientific journal of the security of the production and information exchange space. 17(1):1-18.
[30]Akhlaqpour, Mohammad. (2018). Presenting a machine learning-based intrusion detection system to reduce network attacks based on clustering. National Conference on Modern Research in Electrical, Computer and Information Technology Engineering, (612(7.(
[31] Torabi, Maryam; Mahrooghi, Hamidreza; Aliabadi, Subhan; Amin Toosi, Haleh. (2019). Design and Implementation of a Trust-Based Framework for Mobile Traffic Detection in the Network. Electronic Industries Quarterly, 10(4):69-84.
[32]Anurag Bhardwaj, Ritu Tyagi, Neha Sharma, Akhilendra Khare, Manbir Singh Punia, Vikash Kumar Garg.(2022).Network intrusion detection in software defined networking with self-organized constraint-based intelligent learning framework,Measurement: Sensors, 24(100580):2665-2674.
[33] Abdulsalam O. Alzahrani, Mohammed J. F. Alenazi.(2022).ML-IDSDN: Machine learning based intrusion detection system for software-defined network, Deanship of Scientific Research, King Saud University.35(1):37-56.

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 *Revised: 15-03-2025* *Accepted: 10-04-2025*

[34] Oqbah Ghassan Abbas;Khaldoun KhorzomMohammed Assora.(2020).achine Learning based Intrusion Detection System for Software Defined Networks,International Journal of Engineering Research & Technology (IJERT). 9(9):2278-2281.
[35] A. Abubakar and B. Pranggono,.(2017).Machine learning based intrusion detection system for software defined networks," 2017 Seventh International Conference on Emerging Security Technologies (EST), Canterbury, UK, 138-143,
[36] Poularakis, Konstantinos, George Iosifidis, and Leandros Tassiulas.(2018).SDN-enabled tactical ad hoc networks: Extending programmable control to the edge", IEEE Communications Magazine 56(15): 132-138.
[37] Manso, P.; Moura, J.; Serrão, C.(2019). SDN-Based Intrusion Detection System for Early Detection and Mitigation of DDoS Attacks. Information, 10, 106.
[38] Sarkar, S.K.; Roy, S.; Alsentzer, E.; McDermott, M.B.A.; Falck, F.; Bica, I.; Adams, G.; Pfohl, S.; Hyland, S.L. (2020).Machine Learning for Health (ML4H): Advancing Healthcare for All. Proc. Mach. Learn. Res. 136, 1–11
[39]B. A. A. Nunes, M. Mendonca, X. Nguyen, K. Obraczka, and T. Turletti, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," IEEE Communications Surveys & Tutorials, vol. 16, pp. 1617-1634, 2014.
[40]F. Hu, Q. Hao, and K. Bao, "A Survey on Software-Defined Network and OpenFlow: From Concept to Implementation," IEEE Communications Surveys & Tutorials, vol. 16, pp. 2181-2206, 2014.
[41]W. Xia, Y. Wen, C. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking. IEEE COMMUNICATION SURVEYS & TUTORIALS, 17 (1), 115–124," ed, 2015.
[42]H. Yang and Y. Kim, "SDN-based distributed mobility management," in 2016 international conference on information networking (ICOIN), 2016: IEEE, pp. 337-342.
[43]M.-J. L. Jake Ryan , Risto Miikkulainen, "Intrusion Detectio with Neural Networks", Texas University, 1997.
[44]I. Z. Bholebawa, R. K. Jha, and U. D. Dalal, "Performance analysis of proposed network architecture: OpenFlow vs. traditional network," International Journal of Computer Science and Information Security, vol. 14, no. 3, p. 30, 2016.
[45]H. Zhou et al., "Improving QoS in SDN with lossless multi-domain reconfigurations," in 2015 IEEE 23rd International Symposium on Quality of Service (IWQoS), 2015: IEEE, pp. 77-78.
[46]M. Karakus and A. Durresi, "Quality of service (QoS) in software defined networking (SDN): A survey," Journal of Network and Computer Applications, vol. 80, pp. 200-218, 2017.
[47]D. Li, S. Wang, K. Zhu, and S. Xia, "A survey of network update in SDN," Frontiers of Computer Science, vol. 11, no. 1, pp. 4-12, 2017.
[48]N. Dorsch, F. Kurtz, H. Georg, C. Hägerling, and C. Wietfeld, "Software-defined networking for smart grid communications: Applications, challenges and advantages," in 2014 IEEE international conference on smart grid communications (SmartGridComm), 2014: IEEE, pp. 422-427.

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 *Revised: 15-03-2025* *Accepted: 10-04-2025*

[49]S. E. Abdel-Rahman Hedar & Mohamed Adel Omar & Adel A. Sewisy, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2015 16th IEEE/ACIS International Conference on, 06 August 2015.
[50]S. Scott-Hayward, G. O'Callaghan, and S. Sezer, "SDN security: A survey," in 2013 IEEE SDN For Future Networks and Services (SDN4FNS), 2013: IEEE, pp. 1-7.
[51]T. A. Assegie and P. S. Nair, "A review on software defined network security risks and challenges," Telkomnika, vol. 17, no. 6, 2019.
[52]X. Wen, Y. Chen, C. Hu, C. Shi, and Y. Wang, "Towards a secure controller platform for openflow applications," in Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking, 2013, pp. 171-172.
[53]K. K. Karmakar, V. Varadharajan, and U. Tupakula, "Mitigating attacks in software defined networks," Cluster Computing, vol. 22, no. 4, pp. 1143-1157, 2019.
[54]S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "Avant-guard: Scalable and vigilant switch flow management in software-defined networks," in Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, 2013, pp. 413-424.
[55]C. Yoon et al., "Flow wars: Systemizing the attack surface and defenses in software-defined networks," IEEE/ACM Transactions on Networking, vol. 25, no. 6, pp. 3514-3530, 2017.
[56]S. Shin et al., "Rosemary: A robust, secure, and high-performance network operating system," in Proceedings of the 2014 ACM SIGSAC conference on computer and communications security, 2014, pp. 78-89.
[57]K. Benzekki, A. El Fergougui, and A. Elbelrhiti Elalaoui, "Software-defined networking (SDN): a survey," Security and communication networks, vol. 9, no. 18, pp. 5803-5833, 2016.
[58]R. Masoudi and A. Ghaffari, "Software defined networks: A survey," Journal of Network and computer Applications, vol. 67, pp. 1-25, 2016.
[59]T. Koponen, M. Casado, N. Gude, and J. Stribling, "Distributed control platform for large-scale production networks," ed: Google Patents, 2014.
[60]S. Singh and R. K. Jha, "A survey on software defined networking: Architecture for next generation network," Journal of Network and Systems Management, vol. 25, no. 2, pp. 321-374, 2017.
[61]D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," Proceedings of the IEEE, vol. 103, no. 1, pp. 14-76, 2014.