REPUBLIC OF TURKEY

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Information Technologies

DDOS ATTACKS DETECTION USING MACHINE LEARNING MODELS IN SDNS

Mohammed AL-ADEMI

Master of Science

Supervisor

Prof. Dr. Osman Nuri UÇAN

Istanbul, 2021

DDOS ATTACKS DETECTION USING MACHINE LEARNING MODELS IN SDNS

Mohammed AL-ADEMI

Information Technologies

Master of Science

ALTINBAŞ UNIVERSITY

2021

The thesis titled “DDOS ATTACKS DETECTION USING MACHINE LEARNING MODELS IN SDNS” prepared by MOHAMMED ABDULLAH ABDULWAHID AL-ADEMI and submitted on 03/12 /2021 has been **accepted unanimously of votes** for the degree of Master of Science in Information Technologies.

Prof. Dr. Osman Nuri UÇAN

the Supervisor

Thesis Defense Committee Members:

Prof. Dr. Osman Nuri UÇAN	Faculty of Engineering and Architecture, Altınbaş University	_____
Assist. Prof. Abdullahi Abdu IBRAHIM	Faculty of Engineering and Architecture, Altınbaş University	_____
Prof. Dr. Hasan Huseyin BALIK	Faculty of Electrical and Electronic Engineering, Yıldız Technical University	_____

I hereby declare that this thesis meets all format and submission requirements of a Master’s thesis.

Submission date of the thesis to the Graduate Education Institute: ____/____/____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Mohammed AL-ADEMI

Signature

DEDICATION

I would like to thank my esteemed professor Prof.Dr. Osman Nuri UÇAN who guided me with his valuable help and contributions throughout my studies, guided and helped me while preparing my thesis content, and all my valuable professors in the Computer Engineering Department, from whom I benefited from their valuable experiences.

ACKNOWLEDGEMENTS

Prof. Dr. Osman Nuri UÇAN, my supervisor, deserves my gratitude and acknowledgement. I owe him a great debt of gratitude for his wise counsel during the completion of my thesis. I value his extensive expertise, enthusiasm for the study process, and readiness to assist.

I appreciate my parents', brothers', sisters', and friends' efforts, and I express my heartfelt gratitude to my family, especially my mother and fiancée, who have always been there for me with moral support throughout my academic career. They have made incalculable contributions to my education and professional development.

I applaud the endeavors of faculty members, support staff, and my classmates. I'm also grateful for the compassion and collaboration of the school administration, the Turkish government, and the Turkish community for their worm residency.

ABSTRACT

DDOS ATTACKS DETECTION USING MACHINE LEARNING MODELS IN SDNS

Al-Ademi, Mohammed

M.Sc, Information Technologies, Altınbaş University,

Supervisor: Osman Nuri UÇAN

Date: December/2021

Pages: (39)

SDN (Software Defined Networking) has numerous benefits, including management, scalability, and increased efficiency. Furthermore, SDN has unique security challenges, particularly when the controller of it is subject to Distributed Denial of Service (DDoS) assaults. When DDoS assaults are launched against the SDN controller, the process and communication capacities are exceeded. For the attacker packets, the controller creates superfluous flow, causing the switch flow entries to build up and the system to collapse catastrophically. In this work, various methods were used to identify DDoS assaults in SDN. First, SDN was used to collect specific features for the datasets in both ordinary and DDoS network attacks. We picked the features to reduce the algorithms, making them easier to follow, and reduce the training time. Our features were trained and evaluated using logistic regression.

Classification methods based on K-Nearest Neighbors (KNN) and Artificial Neural Networks (ANN). The results of the tests revealed that the Artificial Neural Network (ANN) classifier had the best accuracy rate (98.9 percent) in detecting DDoS attacks. The findings show that machine learning can improve detection of DDoS assaults in SDN while also reducing loads of the processing and delays.

Keywords: AI Algorithms, DDoS Attacks, Feature Selection, SDN.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

LIST OF ABBREVIATIONS

AI	:	Artificial Intelligence
DDOS	:	Denial-of-Service Attack
SDN	:	Software-defined Networking
ANN		Artificial Neural Network
SVM		Support Vector Machine
KNN	:	K-Nearest Neighbors
APIs	:	Application Programming Interface
ICMP	:	Internet Control Message Protocol

1. INTRODUCTION

Conventional network systems were unable to meet specific criteria, such as virtualization, cloud computing, dynamic management, high connection speed, accessibility and high bandwidth. As a result, Software Defined Networking (SDN), with a customizable, can be programmed, and changeable design, has appeared as a promising substitute for conventional networks [1]. Control, data, and application planes make up the SDN architecture. Devices like as routers and switches are deployed in data layer. The control plane is in responsible for programming and managing this aircraft [2]. The data plane communication equipment is administered by the control aircraft. The control unit is placed on this plane, which acts as networks brain. Packet transmission is performed on the data plane according to controller rules. The application plane connects by a controller with the network infrastructure. Applications utilizing Southbound and Northbound interfaces program network devices. By use of the northbound interface, the controller interacts to apps at the application layer, and a southbound interface, it talks to data layer devices. The OpenFlow Standard is the most widely utilized for this type of connection. Since control logic is taken from and central to local devices, SDN is organized and optimized dynamics from a single location [4]. Although some security dangers in computer networks are common, SDN has added security threats of its own. At least seven attack vectors using SDN [5] are discovered. SDN is most at risk from DDoS assaults, which cause widespread denial-of-service (DoS) damage. SDN controllers can be in the OpenFlow switch's routing table, or they can be in the switch's flow table, together with the controller. DDoS attacks might target the controller because of the open communication connection between it and the data plane. Massive amounts of traffic are being used to launch DDoS attacks against the OpenFlow switch in the data plane. When an OpenFlow switch fails to match a packet with a flow input (miss flow), the packet is placed in the flow buffer. After that, the controller is sent the packet-in message so that it may create a new rule. When the controller sources (memory, CPU, bandwidth, etc.) go down, the Network goes down as well. The controller's communication connection capacity is also a factor, as is the OpenFlow switch's ability to handle attack traffic. This has a substantial impact on network performance [6]. The network device flow table makes the data plane vulnerable to DDoS attacks. Packets are delivered from unknown sources to the switch in such DDoS assaults. The controller sends them to the switch flow table and writes rule for these coming packets.

After a while, the flow table capacity is fulfilled. As such, it is impossible to write new rules to the packets and flow table. Furthermore, when you have the buffer capacity filled with the assault, incoming packets automatically fall [7]. The switches of OpenFlow handle packets with the controller only entering flow. Therefore, it is more challenging to decrease DDoS assaults on SDN than conventional networks systems. Techniques of the Machine learning can deliver more dynamic, efficient, and intelligent networks systems administration, security. The detection of DDoS assaults is extremely crucial in order to guarantee network security to take the required steps promptly. SDN Flow data may lead to a smart network, which can learn and work with machine-based DDoS attack detection systems that are incorporated into the structures of SDN. Moreover, SDN can be a reference model for the creation of a safe structure in research that provides for 5G network integration with an integrated machine learning application. There are five sections in this study. The second section offers a brief analysis of DDoS and OpenFlow controls assaults as well as SDN architecture and features and security breakdowns. DDoS attacks are also included. Under Materials and Methods, the third part presents detailed steps of the process for using methods of feature selection and models of machine learning as well as investigational setup and SDN data collecting. In section four, the outcomes of the experiments have been analyzed and the implications have been drawn. The final observations of the investigation are provided in the last portion of the study.

2. LITERATURE REVIEW

The associated research is discussed from two different viewpoints, notably DDoS assaults on the OpenFlow switch and controller.

2.1 DDOS ATTACKS ON THE CONTROLLER

The controller, which acts as the SDN architecture's brain, assigned control responsibilities to the switch. The controller makes it simple to incorporate the network system's rules at the parent level. The controller can insert additional rules and modify the current rules to the transmission devices. It can make these modifications over a protected channel using the OpenFlow protocol by connecting with transmission devices. This channel ensures continuity and unity of data transmission. The link between the transmission devices and controller breaks if this channel breaks. The target for DDoS assaults is SDN architecture. As an attacker attacks the network, as illustrated in Figure 1 there are three major targets: to consume controller source, to fill the channel bandwidth between the switch and the controller and to load flow tables with needless flows into the switch. The invader sends many packets to the OpenFlow switch through zombie in DDoS assaults against the controller. The controller has trouble telling the difference between normal traffic and traffic delivered by the attacker.

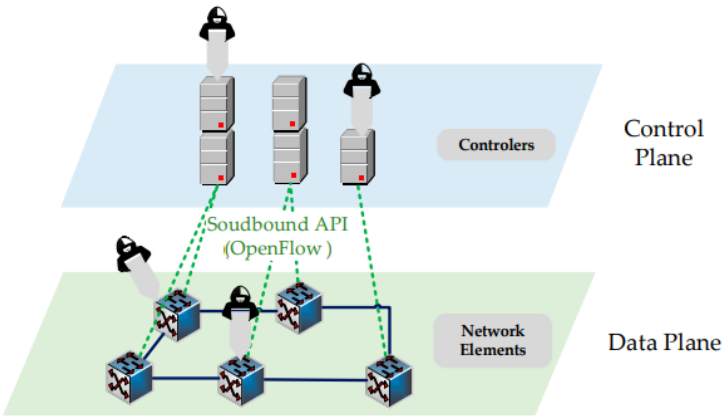


Figure 2.1: Main targets of (DDoS) [1]

When using OpenFlow, the switch will look at the packet header and make sure it matches what's expected in the flow input (target IP address, target IP address, source IP address, target

port, source port, etc.). Using the OpenFlow Protocol (OFPT) PACKET IN, the packet can be delivered to the controller without waiting for a match to be found. Packet header message in a flux request. The controller will get the OFPT FLOW MOD message. In the packet flow table, the message includes the operation to be applied to the packet and the flow timeout [1]. More packets sent to the controller means that resources (bandwidth, memory, and CPU) are being exhausted faster, which means that when more packets enter the network, the controller can no longer handle them. Current DDoS-prevention methods, according to Alshamrani et al. [2] are ineffective. As a result, SDN-related misconduct and novel flow assaults have been looked into. As part of the assault, they acquired traffic data from transmission devices on a data plane, which will be utilized in the SDN architecture to respond to unexpected traffic fluctuations using classification algorithms for machine learning. Messages from the controller to transmission equipment served as the basis for the attack. Classification methods such as SVM, J48, and NB were all supported. DDoS assaults were identified by monitoring the arrival rate of packets during an attack by Latah and Taker [3]. An inspection unit using SVM was triggered to identify DDoS flooding assaults if any packets exceeded the preset threshold. A Recurrent Neural Network (RNN) model presented by Li et al. [4] covers the SDN network topology layer by layer to detect and prevent DDoS attachments. Because it is meant to identify DDoS assaults in real time and has a high precision block, the technique may not function as well in networks with more controllers and larger sizes. Disrupting the coordinated activities of the controllers, as recommended, might impair the network's operation.

2.2 THE OPENFLOW SWITCH AT THE DDOS ATTACK

The flow table and OpenFlow system switch are considered to be the key objectives since they incorporate access controls, transmission information, and administrative information [5]. The initial objective of attacker is to disturb the network's functioning via physical or virtual illegal Internet access.

Because of the limited storage space, the OpenFlow switch is unable to maintain all of the rules for each flow. Because packets come from an unknown source, the switch must implement additional rules to deal with them. Packets from an unknown source may be sent in rapid succession, prompting the controller to login and submit routing rules to the routing table.

Shortly after it is created, the flux table runs out of storage capacity. There isn't place for another rule in the flow table. So the legal transmission of traffic is over. Figure 1.1 shows a DDoS attack on a memory buffer. A reactive cache storage device receives the packet from the input port and sends it to the flow cache memory [5].

The flow table will be searched for a match for the new arrival packets. Once a match is found, the packet is sent to the cache exit port. Assuming this is the case, the controller receives the control channel and the message Packet In and replies by assigning the hard timeout and idle timeout to an OFPT FLOW MOD message, which explains the rules needed for the packet and how long the rules remain in force. Incoming packets are handled fast because controller rules are stored in the cache table memory and sent to the switch when they arrive. Unfortunately, the switch responsible for mitigating DDoS attacks was left unguarded. To protect the switch, the controller must respond to a high number of packets sent by malicious nodes [7]. Legitimate users begin rejecting packets when malicious node packages have finished filling the buffer [8]. Ye et al. [9] utilized the controller to gather network traffic data from data plane transfer devices. To identify DDoS assaults with SVM, six distinctive features associated to DDoS attacks were culled from the switch flow table. The detection accuracy rate was quite good. Despite this, the test accuracy rate for the ICMP attack flow was determined to be poor. Xue et al. [10] noted that extra security considerations are required since the SDN controller handles many data plane switches. They argue that current equipment and software, which have not yet been switched to the SDN architecture, cannot provide adequate security. Particularly the DDoS assault on data level switches threatens the SDN architecture of the network. To identify DDoS attacks, CV-GA utilizes SVM-optimized C and G parameters with a cross-validation genetic algorithm (CV-GA). Nanda et al. [11] advocated the use of machine learning approaches to discover potentially dangerous connections and suitable targets for reducing security concerns by training on historical network attack data. Some of the approaches employed were Naive Bayes, Decision Table (DT), Bayesian Network, and C4.5. Dishonest operators might be identified on the data plane by evaluating machine learning techniques, according to the theory. It is important for the network's efficiency and reliability that the SDN controller can rapidly and efficiently set new rules to avert an attack. SELF-ORGANIZING maps (SOM) machine learning methods, Learning Vector Quantization (LVQ1), and other upgraded variants were used by Jankowski and Amanowicz[12] to assess and analyze hazardous behaviors from a data plane (M-LVQ1) using

Multipass Learning Vector Quantization (M-LVQ1). When compared to SOM, M-SOM, LVQ1, and M-LVQ1, H-LVQ1 exhibited encouraging results. Findings that are encouraging to hear about. In their study, Banerjee and Chakraborty [13] went through two stages of analysis. When trying to figure out if the attack was a hoax, machine learning techniques like Naive Bayes and K-Nearest Neighbors were employed. Three-way handshakes were used to identify the assailant. By developing an access control list, we were able to halt the intruder (ACL). Cognitive switching, according to Mowla et al [14], might be used to detect and mitigate DDoS attacks on content delivery networks. Mehdi and his associates in order to avoid fresh floods, identify and combat all future DDoS assaults, traffic categorization was based on SVM [15] and logistical regression techniques, and these classifications resulted in the establishment of security rules for OpenFlow switches [16]. DDoS assaults against SDN architectures are the subject of this study, which also includes attack selection techniques for learning machine models. A high-fertility machine learning (ML) system for SDN architecture is being developed to detect DDoS assaults [17]. Assaulting the SDN controller and switches on the data plane with DDoS attacks must be detected in order for the network to continue operating for traffic to be detected [18]. To avoid a crash [19], if the controller's traffic is recognized on the controller, new rules will be written into the switch flow chart on the data plane. In terms of averting an attack, this offers a huge benefit. DDoS assaults may be detected via function selection approaches and machine learning models, which we suggest employing. To our knowledge, this approach will greatly improve the ability to identify DDoS assaults [20].

3. MATERIALS AND METHODS

When compared to natural intelligence (EI) shown by humans and other species, artificial intelligence AI is the sort of intelligence produced by computers. Whenever a computer duplicates the "cognitive" processes that people equate with human brains, like "learning" and "problem solving," the phrase "artificial intelligence" is used. In other terms, artificial intelligence is a computer-controlled robot's capacity to perform tasks that are comparable to those performed by intelligent organisms.

3.1 MACHINE LEARNING

Machine learning is a kind of artificial intelligence that allows computers to be able to learn and grow on their own, without the need for special programming. To put it another way, ML is concerned with the creation of software applications that can collect massive datasets on their own. It is a subfield of computer science that employs methods [21]. A person's learning journey begins with observations or data, such as personal stories, firsthand knowledge, and guidance on how to search for patterns in data and use those patterns to make better judgments moving forward. The main objective is for machines to learn within their own and modify their behavior properly without the need for human intervention [22]. machine learning have three types of algorithms: reinforcement, unsupervised and supervised.

3.1.1 Algorithms of Supervised Learning

supervised learning is a ML function that translates an intake to an outcome associated with the input pairs that have been sampled [23].

For intake values given to the system throughout supervised learning, output values are also provided. To create the required outputs for the supplied inputs, the network changes its own weights. The new weights of the network are ordered according to the margin of error between the network's outputs and the predicted outputs, which is calculated. When determining the confidence interval, the differential between all of the network's outputs and the predicted

outputs is computed, and the error margin for each cell is determined based on this difference. So each cell's weights are updated [24].

3.1.2 Unsupervised Learning Algorithms

The training data is neither categorized nor labeled, unlike supervised learning techniques.

Unsupervised ML algorithms look at unlabeled data, looking for similarities and attempting to uncover the data's structure. Although the system may not locate the proper output, it may probe the data and infer from datasets to uncover hidden patterns in unlabeled data.

3.1.3 Reinforcement Learning Algorithms

Reinforcement learning algorithms are a sort of training system to create activities and detects faults or rewards in its environment. Trial and error seeking, as well as delayed reward, are quite well components of reinforcement learning algorithms. This method may be used by machines and software agents to dynamically select the optimum behavior in a given scenario and maximize performance. Simple reward feedback is all that is required for the software agent to figure out which action is best. A reinforcement signal is a term used to describe this sort of feedback [25].

3.2 ISOLATION FOREST ALGORITHM

Isolation forests is one of the newest methods for the detection of abnormalities. The algorithm is because there are minimal abnormalities and discrete datasets. These characteristics are susceptible to an isolation mechanism. This is a very convenient and fundamentally different technique than any current one. It allows separation for the detection of abnormalities in frequently used fundamental measures of distance and density as a more effective and efficient instrument. This approach is also a low linear time complexity algorithm and a modest demand of memory. Irrespective of the size of the dataset, it makes a decent model utilizing tiny subsamples with a small number of trees. Typical techniques of learning machines operate more effectively when they're balanced, which means that the data set has the same amount of excellent and bad performance.

3.3 LOCAL OUTLIER FACTOR (LOF) ALGORITHM

The LOF algorithm is an unmonitored detection technique for the exclusion of a particular data point, which determines its local density divergence from its neighbor. They are regarded as elderly specimens with a considerably reduced density compared to their neighbors. The number of acceptable neighbors (parameter n neighbors) is generally:

1) More than the minimal number of items a set must have in relation to the set so that other objects might have local outliers.

2) is less than the maximum number. such information is usually not available and taking n_neighbors= 20 works well in general.

3.4 DECISION TREES

It is a strategy used by corporate management to analyze numerous decision points connected to repeated chance events. It may help define preferences, risks, rewards, and objectives, and it can be used to a variety of significant investment sectors.

3.5 RANDOM FORESTS

Supervised learning techniques such as Random Forest are used under supervision. As the name implies, it produces a forest at random. Many of the decision trees in his "forest" were trained by utilizing the "bagging" approach. The fundamental idea behind the bagging strategy is that combining several learning models improves the final product. To put it another way, random forest combines several decision trees to generate a prediction that is considerably more precise and reliable. Random forest has a significant edge over most other machine learning techniques. It may be used to solve problems with classification or regression.

Random forest categorization will be addressed in this part. Figure 2.1 depicts how two trees in a random forest would appear:

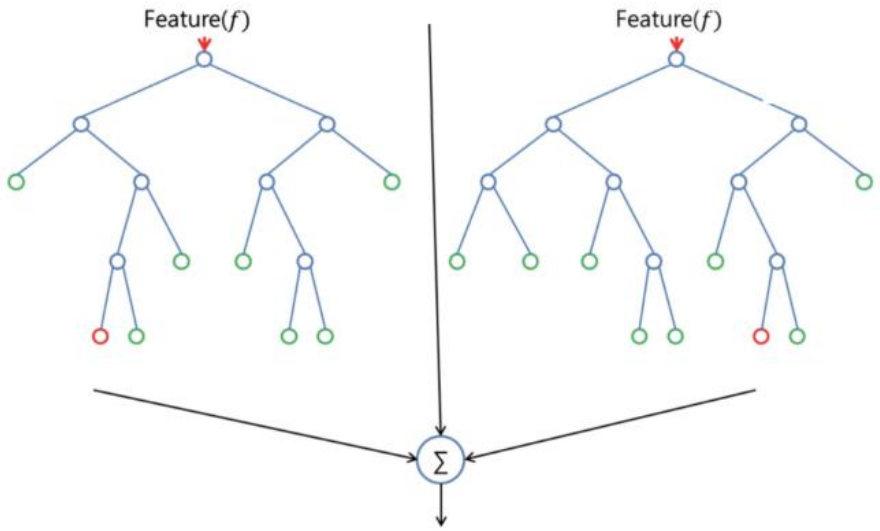


Figure 3.1: TREE of Random Forest [19]

A decision tree or bagging classifier's hyper parameters are quite similar to those of Random Forest. As a result, a decision tree and a bagging classifier are no longer necessary. The Random Forest regression, as previously indicated, may be used for Random Forest regression jobs. Using Random Forest, the model's trees grow in complexity while also becoming more unpredictable. When dividing a node, it doesn't seek for the most important characteristic, but rather the best feature from a random selection of options. Thus, there is a wide range of options, which usually results in a better model.

As a result, in Random Forest, the process of splitting a node only takes a random selection of attributes into account. Using random threshold values for each feature instead of searching for the best feasible thresholds will increase the randomness of the trees (like a regular decision tree).

3.6 ARTIFICIAL NEURAL NETWORKS

The first neural network studies and the developed models will be explained in this section as they form the basis for the models developed later.

Artificial neural networks are computer systems designed to accomplish skills similar to those of the human brain, such as deriving new knowledge, generating new information, and finding new

information through learning, without the need for human intervention. Traditional programming approaches make realizing these skills challenging, if not impossible. As a result, artificial neural networks may be defined as a field of computer science that deals with adaptive information processing for situations that are difficult or impossible to program [26].

Warren McCulloch, a neurologist, and Walter Pitts, a mathematician, published A Logical Calculus of Ideas Immanent in Nervous Activity in 1943, which was the first artificial neural network model. The table below [27] lists the nomenclature for the nervous system parts in the Artificial Neural Network model. The nervous system's cells (A), their interconnections (C), and their artificial neural network equivalents (B, D) are shown.

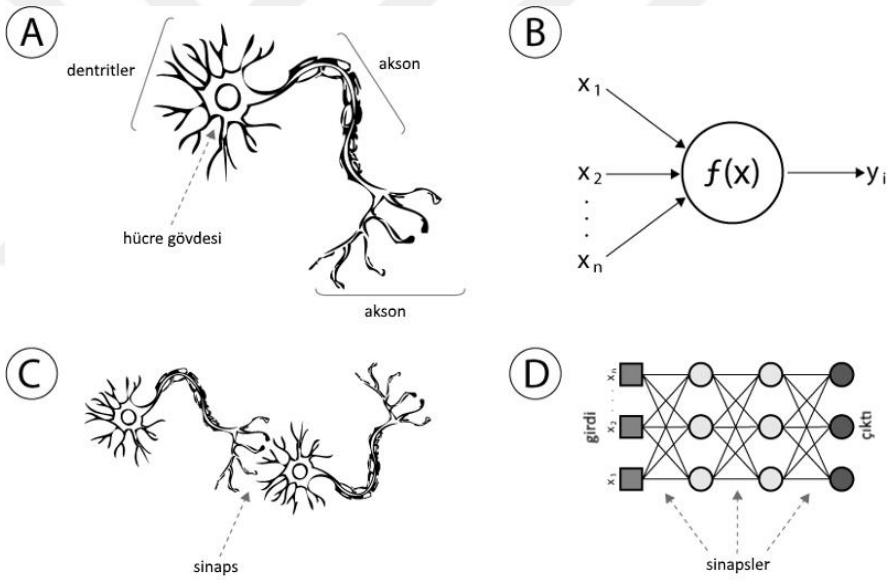


Figure 3.2: A) Nerve cell structure, B) Artificial nerve cell structure, C) Interaction of nerve cells with each other, D) Interaction of artificial nerve cells with each other [26]

Although artificial neural network models and statistical models are similar or identical, the terminologies used differ. The following table shows some terms denoting the same concepts in some neural networks and statistical terminologies.

The characteristics of artificial neural networks that work differently from traditional programming methods are as follows [28].

• Nonlinearity

• Fault tolerance

• Input-output mapping

• Neurobiological analogy

• Adaptability

• Evidence answer

• Contextual information

• Uniformity of analysis and design

3.6.1 Artificial Neuron Structure

ANN have artificial nerve cells in the same way as biological neural networks do. In engineering research, artificial nerve cells are referred to as "process elements." There are five basic elements in each process element [29].These:

• Addition function

• Inputs

• Weights

• Activation function

• Output

In Figure 2.3, G1, G2, G3,.. GN shows the inputs entering the cells, A1, A2, A3,.. AN shows the weight of each input. The aggregation function, which shows how the inputs and their weights will be processed, is shown in the circle. The result of the addition function enters the activation function shown in the square, and the result of this activation function is output to the external environment.

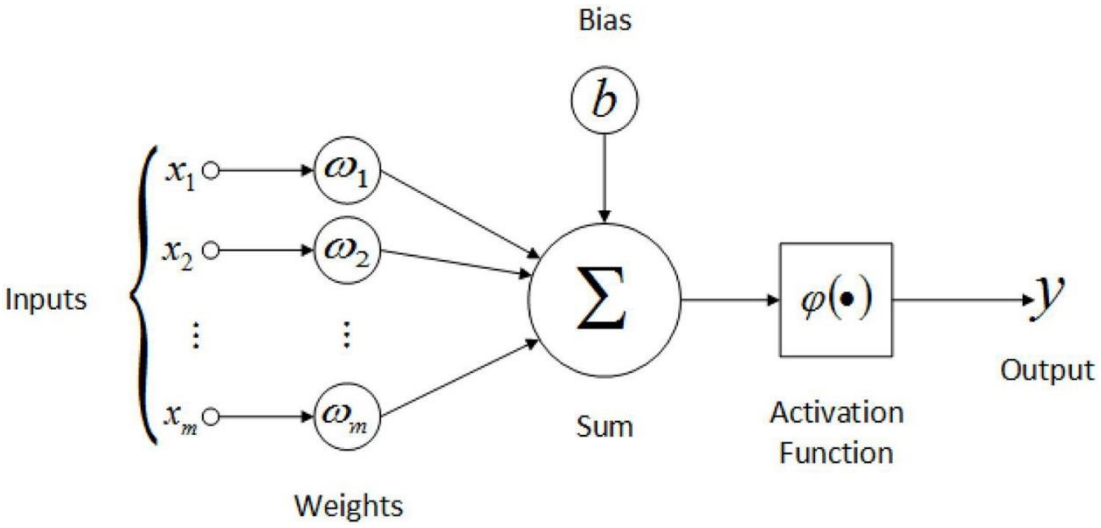


Figure 3.3: Structure of artificial neuron [29]

The information entered from the inputs of the artificial neuron is transmitted to the results according to the elements in the cell and the operations performed.

3.6.1.1 Inputs

It is the information coming to an artificial nerve cell (process element) from the outside world are established by the network examples which is asked to learn. The artificial neuron can receive information from the outside world, as well as from other cells or from itself [30].

3.6.1.2 Weights

Weights indicate the relevance of artificial cell information and the cell impact. The effect of G1 on cell in Figure 2.3 indicates A1. It doesn't matter or not since the weights are huge or tiny. A zero weight may be the network's most significant occurrence. Negative values are not irrelevant. Plus, or minus hence shows if it is a positive or negative effect. A zero shows no action. Variable or constant weights are possible [31].

3.6.1.3 Activation Function

Learning and understanding complicated, nonlinear functional pairings is critical for an artificial neural network. The neural network gains nonlinear characteristics as a result of them. A node's output signal can be used as an input signal for another node in the neural network. In artificial neural networks, the sigmoid function is typically utilized as an activated function for the Multilayer Perceptron model, which is the most widely used function in today's world [32].

3.6.1.4 Cell Output

Cell output is the value that is formed as a result of certain processes of the values of the inputs with the activation function. This value can be given to the cells as an input and can be used directly [33].

3.6.2 Structure of Artificial Neural Network

Layers are common in artificial neural networks. The layers are group of cells having "activation function" that are linked together. The "input layer," which is connected to one or more "intermediate layers," where the real processing is done by weighted connections, receives the examples supplied to the neural network. The result is subsequently sent to the "output layer" by the intermediate levels. [34]

Input layer: Nerve cells in this layer receive information from the outside world and transmit it to the intermediate layers. There may or may not be information processing in the input layers.

Intermediate layers: They are responsible for processing the information coming from the input layer according to the model and sending it to the output layer. While there may be no middleware in a network, there can be one or more middleware.

Output layer: The neurons in this layer process the information coming from the middle layer. They produce the output that the input set (sample) should produce and send the output to the outside world.

In Figure 2.4, the relationship of the artificial neural network layers with each other is shown in a basic way by guiding the arrows.

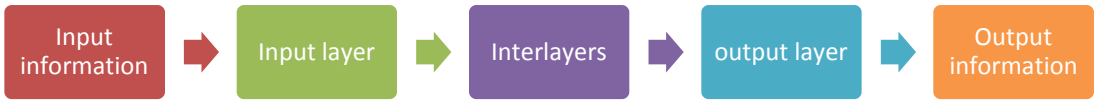


Figure 3.4: Relationship of artificial neural network layers with each other [34]

The circles shown in Figure 2.5 show the nerve cells and the lines between the cells show the connections of the network. There are parallel elements in each layer. Nerve cells and their connections form an artificial neural network.

The weight values of these connections are determined during learning.

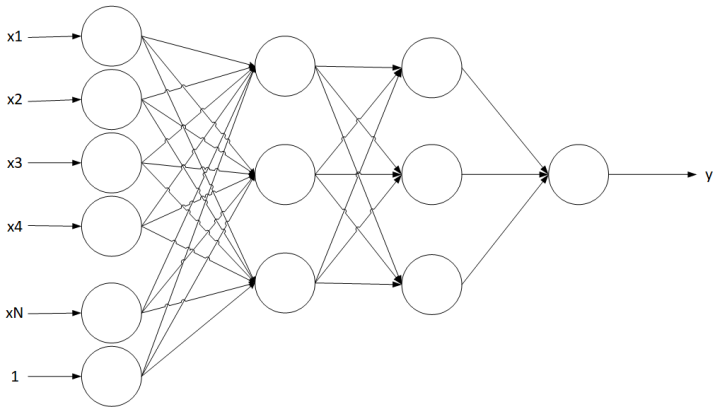


Figure 3.5: Neural Network Example [34]

The most basic of artificial neural networks from the past to the present are mentioned later.

3.6.2.1 Single layer sensors

The input and output layers are the sole layers in single-layer sensors. There are one or more inputs and outputs in every network. All input units are linked to the output units. Every connection carries a certain amount of weight. The network in Figure 2.6 is made up of two inputs and one output. Within these networks, there is also an input threshold that prevents zero values and, as a result, the output of process components. The value of the threshold entry is always 1 [35].

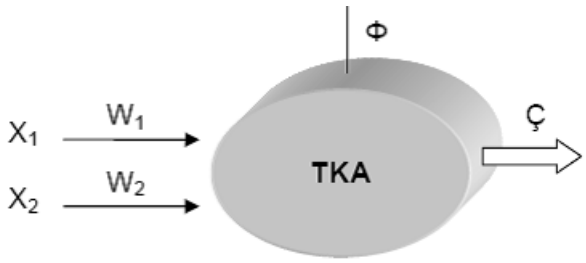


Figure 3.6: Single layer neural network [35]

The network's output is the total of the scaled input values multiplied by the cutoff input (net input). By sending this input via an activation function, the network's output is computed.

3.6.2.2 Simple Sensors

The lightest conceivable neural network is the basic perceptron, which was created in 1957 by Frank Rosenblatt at the Cornell Aeronautical Laboratory. A single nerve cell makes up these structures. One or even more inputs, a process component, and a single output make up a basic sensor. The "feed forward" pattern is followed by a basic sensor. In other words, the inputs are delivered to the neuron, which processes them and generates an output. This implies that a network of neurons is scanned from left to right in Figure 2.7 [16].

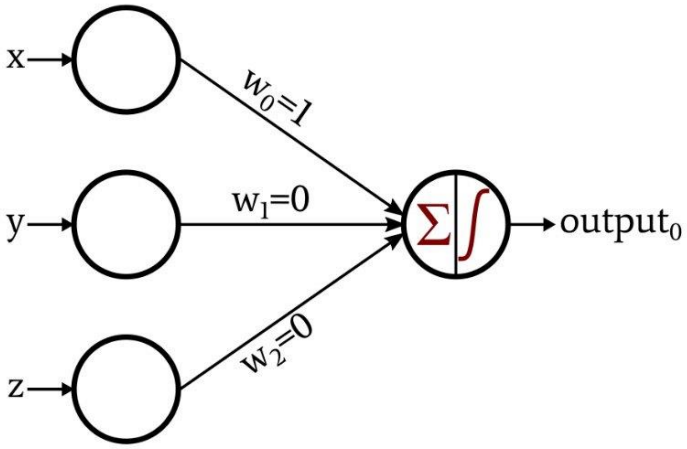


Figure 3.7: Simple sensors [16]

3.6.2.3 Adaline

In 1959 Stanford University's Bernard Widrow created an adaptable linear element called adaline, based on basic neuronal components. It is utilized for a wide array of applications such as speech recognition, character identification, weater provision and adaptive control, with its two layer 'madaline' (multi Adaline) version. Adaline was further improved to generate a continuous output instead of a discontinuous output. In order to prevent echos on telephone lines, Widrow utilized the adaptive linear element method. Mainly the beginning of ANNs to be applied to a real problem [36]. In the structure shown in Figure 2.4, each of the circles represents a separate process element. Adaline structure consists of multiple inputs and one output.

3.6.2.4 Multilevel Sensors

If the relations between the outputs and inputs of the events that an artificial neural network is required to learn are non-linear, it is not possible to learn with the previously mentioned models.

Architecturally, they have a structure in which many nerve cells with non-linear activation functions are hierarchically connected to each other. In general, these structures have two different types of feed the feedback and feedforward [37].

3.6.2.5 Feed Forward Neural Networks

The simplest conceivable neural network is the basic perceptron, which was created in 1957 by Frank Rosenblatt at the Cornell Aeronautical Laboratory. A single nerve cell makes up these structures. One or more inputs, a process element, and a single output make up a basic sensor. The "feed forward" pattern is followed by a basic sensor. In other words, the inputs are delivered to the neuron, which processes them and generates an output. This means that a network of neurons is read from left to right in Figure 2.8. [16].

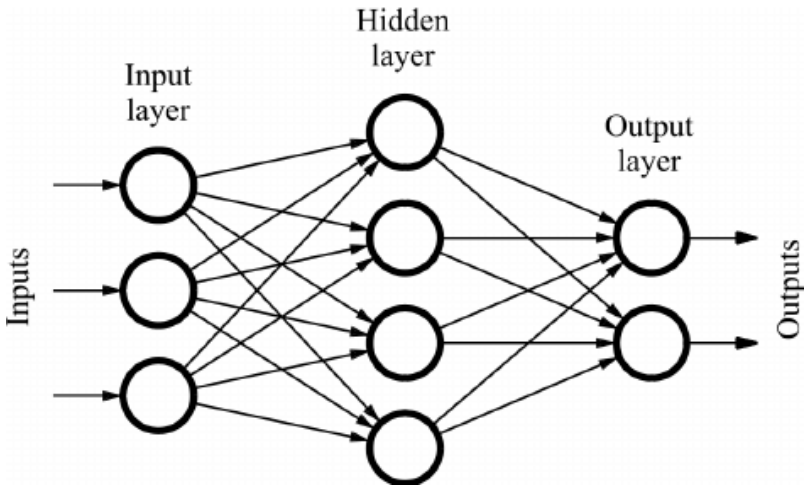


Figure 3.8: Example of feedforward neural network [16]

3.6.2.6 Artificial Neural Networks with Feedback

In a feedback neural network, the results from the output and intermediate levels are transmitted back to the input units or prior intermediate layers. That is why they go forward and backward in time.

Dynamic memories are included in these neural networks, and the instantaneous output reflects both the current and past inputs. Unlike feed forward neural networks, feedback neural networks may process input strings using their internal state (memory). As a result, they may be used for tasks like handwriting recognition [39] and speech recognition [40] [41].

The back propagation method is used to feed feedback artificial neural networks. By adjusting the weights at each node, this method spreads the mistake backwards through the layers [42]. The output calculation and propagation path of the discovered error of a network with two inputs and two outputs seen in Figure 2.6.

The key objective of the back propagation learning approach is to distribute all weights on your network via the error function's effects. This brings the error value down to a bare minimum.

The problem of feedback gradients vanishing in artificial neural networks is a difficulty in artificial neural network back propagation training. At each iteration of the training, each of the neural network's weights receives an update proportionate to the outcome of the error function relative to the current weight. The issue is that the gradient in some situations is too tiny, thereby making it so the weight cannot change. The neural network's training will be terminated if the worst happens. This difficulty was solved by developing a structure known as the long short-term memory.

Long short-term memory (LSTM)

The building blocks for feedback ANN layers are long short-term memory (LSTM) units or blocks. It is a feedback neural network (RNN) that consists of long short-term memory units. An artificial neuron, input gate, output gate, and forget gate make up a basic LSTM unit. The artificial neuron is in charge of "remembering" the value of the long-term memory unit at random times. Gates that regulate the flow of information are found in LSTM blocks. A logistic regression algorithm is used to obtain a value between 0 and 1 for these gates.

Figure 2.9 depicts the construction of a long-term memory cell, which includes input, output, and forget gates. Depending on the time, the configuration of this cell might generate varied effects.

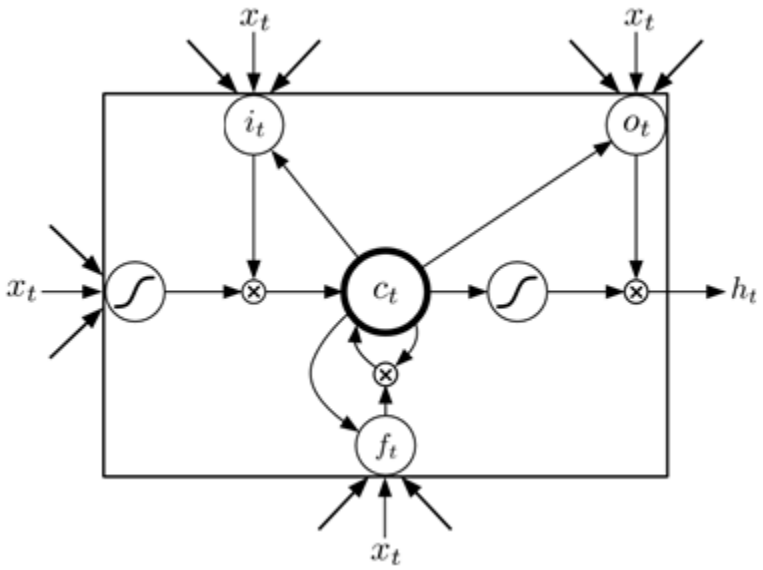


Figure 3.9: Long short-term memory structure [43]

Long short-term memory are intended to help people prevent long-term addiction. It is not something they fight to learn, but it is essentially their natural habit to remember information for lengthy periods of time [43].

3.7 THE STUDY OF ARTIFICIAL NEURAL NETWORKS

The main operating concept of artificial neural networks is to take a collection of inputs and convert them into a set of outputs. The network must be (trained) to create the proper outputs for the inputs presented to it in order to do this. Initially, a vector is created from the input samples before they are transmitted to the network. The network is presented this vector, and it creates the output vector needed for this vector as a result. The network's parameter values are adjusted to generate the proper result. The network developer determines the input and output vector designs, and the samples are gathered in the appropriate format and shown to the network during training.In this study, two of the factors affecting the performance of the artificial intelligence model are discussed. These are the loss function and accuracy rates.

3.7.1 Loss Function

The disparity between the trained model's predictions and the samples that have been trained is represented by loss functions. The distinction between the two domains stems from the objective of generalization. For example, if an artificial intelligence model utilizes the same data all of the time to arrange the artificial neural network, and a new sample, which should be in the same class as a person, produces a completely different result, the loss is significant. The artificial intelligence model is said to have memorized the training set. The goal of machine learning is to reduce the loss of untrained data.

3.7.2 Accuracy Rate

A measure for assessing classification models is accuracy. It's a number that indicates how accurate the artificial intelligence model's inputs and outputs are. This number might range from 0 to 1.

4. APPLICATION

We cover three points of view: DDOS assaults against the OpenFlow switch and the controller in our research. So, in the beginning our controller has to be installed and our OpenFlow switch is changed.

4.1 RYU CONTROLLER DEFINITION

Ryu Controller is an open SDN controller that enhances network agility by making it easier to control and adjust the handling and handling of traffic. With southbound APIs, the SDN Controller sends information from the SDN environment to switches and routers, and with northbound APIs, it sends information from the SDN environment to applications and business logic. NTT supports the Ryu Controller and makes use of it in its cloud data centers. A well-defined set of APIs on the Ryu Controller makes it easy for programmers to create new network management and control apps. This component model enables companies to tailor deployments to suit their unique requirements; developers may change current components fast and easily or create their own components so that the network underlies the changing requirements of their applications.

4.2 MININET

For network simulation purpose, Mininet is used to simulate the software defined network. Mininet enables creating of virtual networks even on desktop PCs or laptops. In Mininet it's possible to create virtual switches and hosts and run them through a process based virtualization through the use of kernel namespace, i.e., Linux network namespaces. The use of kernel namespaces enables it to create a new network namespace for each host created in the virtual network. Mininet starts the controller and open-v Switch processes in virtual machine root namespace by default. Mininet also supports the creation of different topologies for testing of the network experimenting in software-defined networks. Mininet facilitates the rapid prototyping whenever you are working with software-defined networks. It works as a container orchestration tool for emulation of software-defined networks. Starting to use Mininet is as easy as typing the command mn.

Advantages of Mininet:

a) Its fast- takes a few seconds to start a network.

b) It's easy to create custom topologies ranging from a single switch to the data center or back-bone topologies.

c) Can run on a laptop like in this case

d) Its open source

e) Can use customized packet forwarding

Mininet limitations:

a) Since we have a single Linux kernel for each virtualization of hosts, hence software depends on other OS kernels cannot be initiated.

b) Mininet hosts share host file system and process IDs spaces thus need for extra care when running daemons requiring configuration of /etc.

c) Configuration and programming the controller is done by the researcher and not by Mininet.

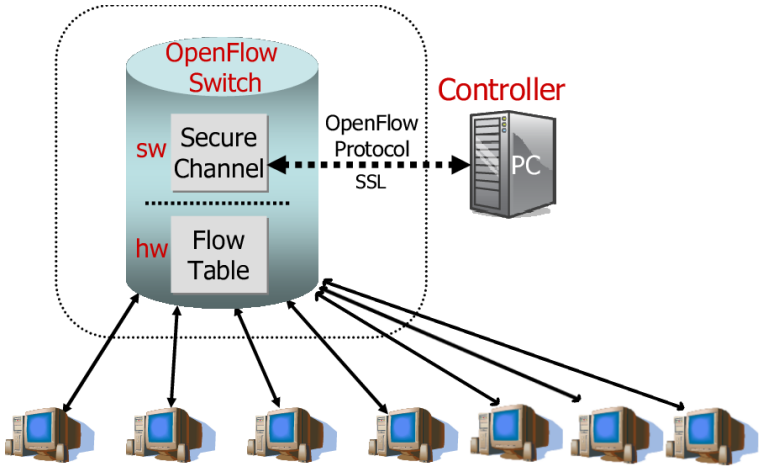


Figure 4.1: OpenFlow Switch Work

4.3 TRAFFIC GENERATIONS

We generated traffic using hping3 which is used to generate four types of traffic which are icmp, tcp, udp, rawid. We made an attack to open flow switch and controller from computer number one to three.

To attack the switch we make an attack from host 1 to host 3 but we send hping flood packets from host 1 to another host which isn't on our network so all request go automatically to the controller because destination host isn't on our network

4.4 DATASET

In the simulation, we generate our dataset which is used in the training of a machine learning model to help the flow based detection module.

Actually our dataset contains 4 other datasets which are shown in the following table

Table 4.1: Dataset Contents

Protocol	Time	Name Of Dataset
tcp	10 Min	attack_tcp
udp	10 Min	attack_udp
icmp	10 Min	attack_icmp
normal	15 Min	flow_normal

Then we combine all of these datasets in one dataset and give it "dataset" name.

The class feature is using to define if there is DDoS attack or not (0 = normal , 1=ddos attack)

Count of total labels in two generalized categories either as an attack or normal is indicated in the following table

Table 4.2: Count of Label Normal-Attack

Label	Count of label
Normal	6736
Attack	13120

Count of labels based on individual attack classes is indicated in the following table

Table 4.3: Count Of Label - Protocol

Label	Count of label
Tcp	4153
Upd	4497
Icmp	4473

4.5 MODEL BUILDING AND EVALUATION

While analyzing the data that is used to design and optimize the model, it is observed that some features do not have a significant contribution to the model in that they even affect the model

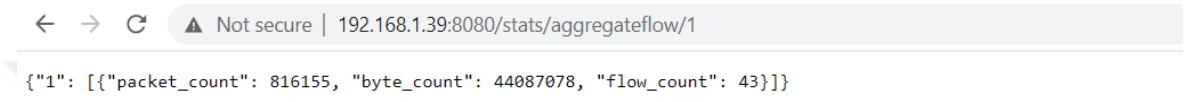
accuracy especially when the insignificant features are many. To avoid inconsistent results that come as a result of using too many features grouped by 2 groups (open flow switch ,controller) showing in the following table

Port Features:

- tx_dropped: packets dropped by RX number
- rx_packets: received packets number.
- rx_crc_err: CRC errors number.
- tx_bytes: received bytes number.
- rx_dropped: packets dropped by TX number.
- port_no: All ports if OFPP_ANY.
- rx_over_err: packets with RX overrun number
- rx_frame_err: frame alignment errors number
- rx_bytes : transmitted bytes number.
- tx_errors: tx_errors
- duration_nsec duration_sec
- collisions: collisions number.
- rx_errors: receive errors number. This is a super-set of more specific receive errors and should be greater than or equal to the sum of all rx_*_err values.
- tx_packets: transmitted packets number.

Flow Switch Features:

- ofs_packet_count : packet count .
- ofs_byte_count : byte count .
- ofs_flow_count : count of records in flow table .



Cpu Features:

- ctrl_cpu_percent : usage of cpu
- ctrl_memory_percent : usage of memory
- ctrl_swap_memory_percent : Ram / swap usage
- etc…

The processes involved in using machine learning models are outlined in the preceding section. Additionally, this section discusses how the dataset's data is gathered and the dataset's features and classifications. During the study, the primary metrics necessary to maintain the SDN architecture's continuity were generated and used to create the dataset's characteristics. Machine learning techniques were used to classify the DDoS assault data.

First, during DDoS attack traffic, special characteristics were derived from the SDN for the dataset. Our specified functionality wanted the model to be simplified, their understanding facilitated and the training time shortened. The Logistic Regression, Artificial Neural Network (ANN) and K-Nearest (KNNs) trained and evaluated our features. The findings indicated the best accuracy rate of DDoS attack detection (98.9 percent) for the Artificial Neural Network (ANN) classifier.

Used algorithms and accuracy is given in below:

Algorithms	**Accuracy**
LogisticRegression	97.24
KNeighborsClassifier	98.48
MLPClassifier (neural_network)	98.91

5. REFERENCES

[1] Kuerban, M.; Tian, Y.; Yang, Q.; Jia, Y.; Huebert, B.; Poss, D. FlowSec: DOS Attack Mitigation Strategy on SDN Controller. In Proceedings of the IEEE International Conference on Networking, Architecture and Storage (NAS), Long Beach, CA, USA, 8–10 August 2016.

[2] Alshamrani, A.; Chowdhary, A.; Pisharody, S.; Lu, D.; Huang, D. A defense system for defeating DDoS attacks in sdn based networks. In Proceedings of the 15th ACM International Symposium on Mobility Management and Wireless Access, Miami Beach, FL, USA, 21–25 November 2017. 10.

[3] Latah, M.; Toker, L. A novel intelligent approach for detecting DoS flooding attacks in software-defined networks. Int. J. Adv. Intell. Inform. 2018, 4, 11–20. [CrossRef]

[4] Li, C.; Wu, Y.; Yuan, X.; Sun, Z.; Wang, W.; Li, X.; Gong, L. Detection and defense of DDoS attack–based on deep learning in OpenFlow-based SDN. Int. J. Commun. Syst. 2018, 31, e3497. [CrossRef]

[5] Li, W.; Meng, W.; Kwok, L.F. A survey on OpenFlow-based Software Defined Networks: Security challenges and countermeasures. J. Netw. Comput. Appl. 2016, 68, 126–139. [CrossRef]

[6] Padmaja, S.; Vetriselvi, V. Mitigation of switch-Dos in software defined network. In Proceedings of the International Conference on Information Communication and Embedded Systems (ICICES), Chennai, India, 25–26 February 2016.

[7] Polat, H.; Polat, O. The effect of DoS attack on ODL and POX SDN controllers. In Proceedings of the 8th International Conference on Information Technology (ICIT), Amman, Jordan, 17–18 May 2017.

[8] Bholebawa, I.Z.; Dalal, U.D. Design and Performance Analysis of OpenFlow-Enabled Network Topologies Using Mininet. Int. J. Comput. Commun. Eng. 2016, 5, 419–429. [CrossRef]

[9] Ye, J.; Cheng, X.; Zhu, J.; Feng, L.; Song, L. A DDoS attack detection method based on SVM in software defined network. Secur. Commun. Netw. 2018, 2018, 1–8. [CrossRef]

[10] Xue, L.; Yuan, D.; Hu, H.; Ran, J.; Li, S. DDoS detection in SDN switches using support vector machine classifier. In Proceedings of the Joint International Mechanical, Electronic and Information Technology Conference (JIMET-15), Chongqing, China, 18–20 December 2015.

[11] Nanda, S.; Zafari, F.; DeCusatis, C.; Wedaa, E.; Yang, B. Predicting network attack patterns in SDN using machine learning approach. In Proceedings of the 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Palo Alto, CA, USA, 7–10 November 2016.

[12] Jankowski, D.; Amanowicz, M. On efficiency of selected machine learning algorithms for intrusion detection in software defined networks. Int. J. Electron. Telecommun. 2016, 62, 247–252. [CrossRef]

[13] Chakraborty, S.; Banerjee, S. Proposed approach to detect distributed denial of service attacks in software defined network using machine learning algorithms. Int. J. Eng. Technol. 2018, 7, 472–476.

[14] Mowla, N.I.; Doh, I.; Chae, K. CSDSM: Cognitive switch-based DDoS sensing and mitigation in SDN-driven CDNi word. Comput. Sci. Inf. Syst. 2018, 15, 163–185. [CrossRef]

[15] Hira, Z.M.; Duncan, F.G. A Review of Feature Selection and Feature Extraction Methods Applied on Microarray Data. Adv. Bioinform. 2015, 2015, 1–13. [CrossRef] [PubMed]

[16] Venkatesh, B.; Anuradha, J. A Review of Feature Selection and Its Methods. Cybern. Inf. Technol. 2019, 19, 3–26. [CrossRef]

[17] Stamp, M. Introduction to Machine Learning with Applications in Information Security; Chapman and Hall/CRC: Boca Raton, FL, USA, 2017.

[18] Somol, P.; Novovicova, J.; Pudil, P. Improving Sequential Feature Selection Methods' Performance by Means of Hybridization. In Proceedings of the IASTED International Conference Advances in Computer Science and Engineering, Sharm El Sheikh, Egypt, 15–17 March 2010.

[19] Mostafaei, H.; Menth, M.; Obaidat, M.S. A Learning Automaton-Based Controller Placement Algorithm for Software-Defined Networks. In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, UAE, 9–13 December 2018.

[20] Li, C.; Wu, Y.; Yuan, X.; Sun, Z.; Wang, W.; Li, X.; Gong, L. Detection and defense of DDoS attack–based on deep learning in OpenFlow-based SDN. Int. J. Commun. Syst. 2018, 31, e3497. [CrossRef]

[21] Expert System Marketing Staff, «Expert System,» [Çevrimiçi]. Available: http://www.expertsystem.com/machine-learning-definition/. [Erişildi: 09 Mayıs 2018].

[22] A. L. Samuel, «Some Studies in Machine Learning Using the Game of Checkers,» IBM Journal of Research and Development, cilt 3, no. 3, 1959.

[23] S. J. Russell ve P. Norvig, Artificial Intelligence: A Modern Approach, 2009.

[24] S. J. Russell ve P. Norvig, Artificial Intelligence: A Modern Approach, 2009.

[25] İ. Çayıroğlu, «İleri algoritma analizi - Yapay sinir ağları,» 2015.

[26] DEVHUNTER, «Yapay Zeka,Robotik ve Sinirbilim,» 11 Temmuz 2018.

[27] E. Öztemel, Yapay sinir ağları, 2 dü., İstanbul: Papatya Yayıncılık, 2006, pp. 23-24,29,59,59,48,49,49,50,54.

[28] M. Uslu.a Internet: http://kod5.org/yapay-sinir-aglari-ysa-nedir/. [Erişildi: 23 Nisan 2018].

[29] R.Kishor, Internet: https://www.scribd.com/doc/55236480/AI-Characteristics-of-Neural-Networks. [Erişildi: 29 Nisan 2018].

[30] N.Şengöz"Internet:http://www.derinogrenme.com/2017/03/04/yapay-sinir-aglari/. [Erişildi: 09 Mayıs 2018].

[31] Internet: http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html. [Erişildi: 09 Mayıs 2018].

[32] Internet: https://veribilimcisi.com/2017/08/13/yapay-sinir-aglari/. [Erişildi: 10 Mayıs 2018].

[33] S. Nayak, "Nature-inspired optimization," Fundamentals of Optimization Techniques with Algorithms, pp. 271–296, 2020, doi: 10.1016/B978-0-12-821126-7.00010-3.

[34] Kıraç, «Türkiye'de faaliyet gösteren ticari bankaların finansal risklerinin yapay sinir ağları ile belirlenmesi,» 2011.

[35] A. Zell, Simulation Neuronaler Netze, Addison-Wesley, 1994, p. 73.

[36] A. Graves, M. Liwicki, S. Fern´andez, R. Bertolami, H. Bunke ve J. Schmidhuber, A Novel Connectionist System for Unconstrained Handwriting Recognition, 2009.

[37] X. Li ve X. Wu, CONSTRUCTING LONG SHORT-TERM MEMORY BASED DEEP RECURRENT NEURAL NETWORKS FOR LARGE VOCABULARY SPEECH RECOGNITION, 2014.

[38] H. Sak, A. Senior ve F. Beaufays, Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling, 2014.

[39] P. J. Werbos, Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences, Harvard University, 1975.

[40] Internet:http://derindelimavi.blogspot.com.tr/2017/03/kerasa-giris-2-lstm.html. [Erişildi: 10 Mayıs 2018].

[41] Internet: https://en.wikipedia.org/wiki/Long_short-term_memory. [Erişildi: 10 Mayıs 2018].

[42] A. Graves, A.-r. Mohamed ve G. Hinton, Speech Recognition with Deep Recurrent Neural Networks, 2013.

[43] N. Le Roux, Y. Bengio ve A. Fitzgibbon, «Improving First and Second-Order Methods by Modeling Uncertainty,» Optimization for Machine Learning, p. 404, 2012.

[44] Google Developers, Internet: https://developers.google.com/machine-learning/crash-course/classification/accuracy. [Erişildi: 10 Mayıs 2018].