
GAZİ ÜNİVERSİTESİ
Gazi
1926

PERFORMING INTRUSION DETECTION IN INTERNET OF THINGS (IOT) BY FEATURE SELECTION WITH MAJORITY VOTING APPROACH

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
GAZİ UNIVERSITY

BY

Hussein SHEET AHMED AHMED

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER
IN

DEPARTMENT OF ELECTRICAL - ELECTRONIC ENGINEERING

JUNE 2023

ETHICAL STATEMENT

I hereby declare that in this thesis study I prepared in accordance with thesis writing rules of Gazi University Graduate School of Natural and Applied Sciences;

- All data, information and documents presented in this thesis have been obtained within the scope of academic rules and ethical conduct,
- All information, documents, assessments and results have been presented in accordance with scientific ethical conduct and moral rules,
- All material used in this thesis that are not original to this work have been fully cited and referenced,
- No change has been made in the data used,
- The work presented in this thesis is original,

or else, I admit all loss of rights to be incurred against me.

Hussein SHEET AHMED AHMED

14/06/2023

PERFORMING INTRUSION DETECTION IN INTERNET OF THINGS (IOT) BY FEATURE SELECTION WITH MAJORITY VOTING APPROACH

(M.Sc. Thesis)

Hussein SHEET AHMED AHMED

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

June 2022

ABSTRACT

The Internet of Things (IoT) is a large and intelligent communication network. The volume of traffic in the IoT network is very high and an intelligent switching network is needed to establish communication. Software-oriented network is a practical technology for switching data in the IoT network. In the software-based network (SDN), the controllers are responsible for programming the network switches. One of the applications of software-based networking is the analysis of network traffic in connected switches. Each switch in the software-based network can also play the role of an intrusion detection system and analyze network traffic. The challenge of software-based network intrusion detection systems is to use all characteristics of network traffic to detect anomalies in network traffic. Failure to choose intelligent features in switches or software-based network controllers, reduces the accuracy and speed of attack detection. In this article, an intelligent intrusion detection system in a software-based network with three group intelligence algorithms is presented. In the proposed system, the controllers use the Harris Hawks Optimization (HHO) Algorithm, the Jellyfish Search Optimizer (JSO) Algorithm and the Whale Optimization Algorithm (WOA) to select the features. In the proposed method, a feature is determined by majority voting and if at least two of the three methods select a feature, then the desired feature is selected. Software network switches reduce the dimensions of network traffic based on network selection characteristics and use multi-layer neural network to classify network traffic. The accuracy, sensitivity, and specificity of the suggested method are equal to 99.67%, 99.54%, and 99.41%, respectively, according to the analysis of the proposed method using the NSL-KDD intrusion detection dataset. The proposed method in detecting attacks on the network is more accurate than HHO, WOA and JSO methods in detecting attacks on the IoT network.

Science Code	:	90542
Key Words	:	Intrusion detection system, software-based network, Internet of Things, feature selection, group intelligence, artificial neural network
Page Number	:	61
Supervisor	:	Prof. Dr. Murat YÜCEL

ÇOĞUNLUK OYLAMA YAKLAŞIMIYLA ÖZELLIK SEÇIMI YAPARAK NESNELERIN İNTERNETINDE (IOT) SALDIRI TESPITININ GERÇEKLEŞTIRILMESI

(Yüksek Lisans Tezi)

Hussein SHEET AHMED AHMED

GAZİ UNIVERSITESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Haziran 2023

ÖZET

Nesnelerin İnterneti (IoT), büyük ve akıllı bir iletişim ağıdır. IoT ağındaki trafik hacmi çok yüksektir ve iletişim kurmak için akıllı bir anahtarlama ağına ihtiyaç vardır. Yazılım odaklı ağ, IoT ağındaki verileri değiştirmek için pratik bir teknolojidir. Yazılım tabanlı ağda (SDN), ağ anahtarlarının programlanmasından denetleyiciler sorumludur. Yazılım tabanlı ağ uygulamalarından biri, bağlı anahtarlardaki ağ trafiğinin analizidir. Yazılım tabanlı ağdaki her anahtar, aynı zamanda bir saldırı tespit sistemi rolü oynayabilir ve ağ trafiğini analiz edebilir. Yazılım tabanlı ağ saldırı tespit sistemlerinin zorluğu, ağ trafiğindeki anormallikleri tespit etmek için ağ trafiğinin tüm özelliklerini kullanmaktır. Anahtarlarda veya yazılım tabanlı ağ denetleyicilerinde akıllı özelliklerin seçilmemesi, saldırı tespitinin doğruluğunu ve hızını azaltır. Bu makalede, üç grup zeka algoritmasına sahip yazılım tabanlı bir ağda akıllı bir saldırı tespit sistemi sunulmaktadır. Önerilen sistemde kontrolörler özellikleri seçmek için Harris Hawks Optimizasyon (HHO) Algoritmasını, Jellyfish Search Optimizer (JSO) Algoritmasını ve Balina Optimizasyon Algoritmasını (WOA) kullanmaktadır. Önerilen yöntemde çoğunluk oylamasıyla bir özellik belirlenir ve üç yöntemden en az ikisi bir özellik seçerse istenen özellik seçilir. Yazılım ağ anahtarları, ağ seçimi özelliklerine dayalı olarak ağ trafiğinin boyutlarını azaltır ve ağ trafiğini sınıflandırmak için çok katmanlı sinir ağı kullanır. Önerilen yöntemin NSL-KDD izinsiz giriş tespit veri seti kullanılarak yapılan analizine göre, önerilen yöntemin doğruluğu, duyarlılığı ve özgüllüğü sırasıyla %99,67, %99,54 ve %99,41'e eşittir. Ağa yapılan saldırıları tespit etmede önerilen yöntem, IoT ağına yapılan saldırıları tespit etmede HHO, WOA ve JSO yöntemlerinden daha isabetlidir.

Bilim Kodu	:	90542
Anahtar Kelimeler	:	Saldırı tespit sistemi, yazılım tabanlı ağ, Nesnelerin İnterneti, özellik seçimi, grup zekası, yapay sinir ağı.
Sayfa Adedi	:	61
Danışman	:	Prof. Dr. Murat YÜCEL

ACKNOWLEDGMENTS

Praise be to ALLAH, the Most Gracious and the Most Merciful, who has been so kind to us during this academic year. Without His guidance, it would not have been possible to complete this endeavor. I want to extend my sincere gratitude to my supervisor, Prof. Dr. Murat YÜCEL, for all of his assistance, direction, counsel, and support throughout this master's thesis project. My late, dear mother, father, sisters, brothers, as well as everyone of my family and friends, deserve my sincere gratitude. I must express my profound gratitude to Gazi University for accepting me amid thousands of applicants and for giving me the best education possible. I'd like to end by thanking the Republic of Turkey and the Government of Iraq. I'm grateful that the Turkish Republic has been my motherland throughout my entire stay.

TABLE OF CONTENTS

Page

ABSTRACT v

ÖZET iv

ACKNOWLEDGMENTS v

TABLE OF CONTENTS vii

LIST OF TABLES ix

LIST OF FIGURES x

LIST OF ABBREVIATIONS xi

1. INTRODUCTION 1

1.2. Problem statement 4

1.3. Motivation and importance of the study 4

1.4. Objective of Research 4

1.5. Organization of the Thesis 5

2. LITERATURE REVIEW 7

2.1. Background 7

2.2. Literature review 7

2.3. Whale Optimization Algorithm (WOA) 15

2.4. Harris Hawks Optimization (HHO) Algorithm 16

3. MATERIAL AND METHODS 23

3.1. The proposed framework 23

3.2. K-NN algorithm 26

3.3. Network traffic preprocessing 26

3.4. Objective function 27

3.5. Feature selection with WOA algorithm 28

Page

3.6. Feature selection with HHO algorithm 31

3.7. Feature selection with JSO algorithm 34

3.8. Binarizing the feature vectors 36

3.9. Voting and classification 38

3.10. Data set 40

4. RESULTS AND DISCUSSION 43

4.1. Analysis 43

4.2. Evaluation indices 43

4.3. Implementation parameters 43

4.4. Tests and evaluations 44

5. CONCLUSION AND FUTURE WORKS 53

5.1. Conclusion 53

5.2. Future work 54

REFERENCES 55

CURRICULUM VITAE 61

LIST OF TABLES

Table **Page**

Table 3.1. Characteristics of NSL-KDD dataset for analysis of the suggested approach 40

Table 4.1. Changes in accuracy, sensitivity and specificity index in suggested approach by increasing the number of hidden layers with the WOA feature selection method 44

Table 4.2. Changes in accuracy, sensitivity, and specificity index in suggested approach by increasing the number of hidden layers with the HHO feature selection method 44

Table 4.3. Changes in accuracy, sensitivity and specificity index in suggested approach by increasing the number of hidden layers with the JSO feature selection method 45

Table 4.4. Changes in accuracy, sensitivity, and specificity index in suggested approach by increasing the number of hidden layers with feature selection method with majority voting mechanism 45

LIST OF FIGURES

Figure **Page**

Figure 2.1. Rotational and spiral movements of the whales around the prey 16

Figure 2.2. Movements of the hawk to search around the prey 17

Figure 2.3. Jellyfish behaviors in JSO algorithm modeling 18

Figure 2.4. Software-defined network layers 19

Figure 3.1. Conceptual model of the proposed method in detecting attacks in SDN network 25

Figure 3.2. Feature selection flowchart with Whale's algorithm in suggested approach 29

Figure 3.3. Feature selection flowchart with HHO algorithm in the proposed method 33

Figure 3.5. Feature selection flowchart with JSO Algorithm in the proposed method 35

Figure 3.6. The performance of mapping functions with a range between zero and one 37

Figure 3.7. Binarizing the feature vector with thresholding 37

Figure 3.8. Voting and classification mechanism in the proposed intrusion detection system 39

Figure 4.1. Comparison of accuracy index of feature selection methods 46

Figure 4.2. Comparison of the sensitivity index of feature selection methods 46

Figure 4.3. Comparison of accuracy index of feature selection methods 47

Figure 4.4. Comparison of attack detection accuracy in the proposed method with feature selection methods 48

Figure 4.5. Comparison of the accuracy index of the proposed method with feature selection methods with optimization approach 50

Figure 4.6. Comparison of the sensitivity index of the proposed method with the feature selection methods with the optimization approach 50

Figure 4.7. STD, MAE, Mean, and process time 51

LIST OF ABBREVIATIONS

The abbreviations used in this study are presented below with their explanations.

Abbreviations	Descriptions
k-NN	K-Nearest Neighbor
ANN	Artificial Neural Network
MSE	Mean Squared Error
GWO	Gray Wolf Optimization
DE	Differential Evolution
PSO	Particle Swarm Optimization
DoS	Denial of Services
NIDS	Network intrusion detection system
DoS	Denial of Services
DDoS	Distributed Denial of Service
NIDS	Network intrusion detection system
HHO	Harris Hawks Optimization
MADP-IIME	malware attack detection protocol in IoT-enabled industrial multimedia environment
LSTM	Long Short-Term Memory
IDS	Intrusion Detection System
RFID	Radio-frequency identification
PSO	particle swarm optimization
BBO	biogeography-based optimization
JSO	Jellyfish Search Optimizer
SDN	Software-defined networks
ONF	Open Networking Foundation
ReLU	rectified linear unit
TP	True Positive
TN	True Negative
FP	False Positives
	False Negative

1. INTRODUCTION

1.1. Background

Today, information and communication technologies are developing and the goal of these technologies is to change the traditional patterns of network communication. IoT is one of the network communication technologies that connect smart objects. The IoT is a huge network of smart objects and sensors around us. The IoT includes all kinds of heterogeneous smart devices and each object has integrated smart sensors. These intelligent and interconnected devices communicate with each other without any human interaction (Omolara et al., 2022). The IoT has various applications such as smart homes, transportation, healthcare, education, industrial production, supply chain management, agriculture, etc. (Jarial, 2022). Despite the numerous advantages of the IoT, this communication network has many challenges. One of the challenges of the IoT is security issues. Objects connected to the IoT network have few computing resources and from this point of view, complex security protocols cannot be implemented on them. The low security of nodes and objects used in the IoT has made them susceptible to all kinds of attacks on the network (Guezzaz et al., 2022). For example, the cameras used in the IoT have weak security and in some cases their default password cannot be changed by the owners. Drones or other smart objects are infected with various types of malware and viruses and become botnets. Botnets are machines or nodes infected with malware that carry out attacks against network services in coordination with other botnets. The aim of attacks such as botnet is to shut down a server due to a flood of false requests (R. Li et al., 2022). Denial of Services (DoS) attacks are an example of attacks carried out by botnets on the network (Kharkwal et al., 2022). Studies show that the total IoT market worldwide in 2020 was worth about 389 billion dollars. It is predicted that the value of the IoT network will reach more than one trillion dollars by 2030. Estimates show that billions of bytes of data are generated from the Internet every second. A network with this high value and high volume of generated traffic is a suitable target for all kinds of attacks and it is necessary to design intelligent security systems for this network (K.-H. Le et al., 2022). The increase of IoT devices to billions has caused the security challenges of this network to increase. Estimates show that in the next few years, the number of objects connected to the IoT network will reach more than 70 billion objects (Huo et al., 2022). The connected objects are easily infected and will play the role of a botnet and a node attacking the network. The security challenges of the IoT are

diverse and multidimensional. Operating system-level attacks include spoofing, low-level Sybil, and vulnerable physical interfaces. At the network level, there are recurrent attacks, sinkhole attacks, and wormhole attacks. Attackers take advantage of this vulnerability and launch Distributed Denial of Service (DDoS) attacks to interfere with Internet services by flooding the target's processing power with Internet traffic (Haque et al., 2022). IoT attacks can be very damaging. For example, the Mirai virus infected more than 65,000 IoT devices in the first 20 hours after its release on August 1, 2016 and the number of systems infected with malware reached more than 300,000 (K.-H. Le et al., 2022)(Harada et al., 2022). A DDoS assault using the Mirai protocol with 620 Gbps of bandwidth was launched against the website of the security consulting company Brian Krebs in September 2016. A DDoS assault with a peak bandwidth of 1.1 terabits per second that was directed at a French cloud service provider is another illustration. A Mirai variant took down hundreds of websites in October 2016, including Twitter, Netflix, Reddit, and GitHub, for several hours (K.-H. Le et al., 2022)(Raju et al., 2021). According to statistics from Security Today and Threat Post, there will be around 31 billion IoT devices in use by 2020, but half of them are vulnerable to serious assaults (K.-H. Le et al., 2022). The mentioned cases all, show how vulnerable the IoT network is to various attacks and significant damage is caused to the network infrastructure. Providing a Network intrusion detection system (NIDS) (Chalé & Bastian, 2022) to the network to detect attacks in the IoT network is important and necessary and this article is provided in this direction.

The software-based network is a packet switching system that consists of controllers and a number of switches connected to a controller. Each controller has a data and control panel and a number of switches are connected to it. In some studies, SDN switches have been used as intrusion detection systems. Deep learning (Hadi & Mohammed, 2022), neural network (Domínguez-Limaico et al., 2023), decision trees (T.-T.-H. Le et al., 2022) and support vector machine (Sokkalingam & Ramakrishnan, 2022) are among the learning methods for intrusion detection in software-based network. These researches analyze network traffic in SDN switches and detect attacks. The challenge of intrusion detection systems in SDN switches to deal with attacks is as follows:

- There is no feature selection or their feature selection is mostly based on statistical methods that are not very accurate.

- Using a feature selection method increases the probability of selecting less important features.
- They do not have a black list to store the addresses of nodes attacking the network.

In this article, a smart method is presented in SDN switches to detect intrusion into the network. In the proposed method, the controller uses three Harris Hawks optimization algorithms (Heidari et al., 2019), Whale optimization algorithm (Mirjalili & Lewis, 2016) and Jellyfish Search Optimizer Algorithm (Chou & Truong, 2021) to select features. In the feature selection phase, a feature is selected by majority voting among three feature selection methods and each switch reduces network traffic based on the features selected by three group intelligence algorithms. In each switch, a multi-layer neural network is trained by the optimal feature vector to detect attack traffic. Each switch has a blacklist of attacking IP nodes, which is exchanged with other switches. The innovations of this article to detect attacks on IoT network are as follows:

- Presenting the binary version of three algorithms HHO, WOA and JSO
- Majority voting for feature selection among three algorithms HHO, WOA and JSO
- Classification of network traffic in SDN switches by combining multilayer neural network and three algorithms HHO, WOA and JSO
- Use of blacklist and blacklist exchange between SDN switches

The purpose of this article is to present an intrusion detection system in the context of SDN architecture and to deal with attacks in the packet-switching phase. Another goal of the paper is to accurately identify attacks with intelligent feature selection. Another goal is to increase the security of the IoT network. The contributions of the authors for this article are as follows:

- Providing an intrusion detection system in SDN switches
- Effective selection of important features of network traffic in intrusion detection with the voting mechanism
- Using three group intelligence algorithms with a majority voting mechanism
- Combination of group intelligence and neural network to detect attacks
- Exchanging and sharing malicious IPs between SDN switches

This article has been prepared and written in several sections. The background for the research is given in the second section. The suggested intrusion detection system is introduced in the third section. In the fourth section, the proposed method is implemented and analyzed. Section 5 discusses the article's conclusions and suggestions for future work.

1.2. Problem statement

One of the major challenges in the IoT network is network attacks and network infrastructure. IoT attacks can be carried out by botnets or malware-infected nodes. In this method of attack, there is a main node in the network that the hacker uses to prepare the attack. In this method of attack, the hacker infects a large number of network nodes with malware or a virus, and these machines in the network inadvertently attack network servers. In this way, attacks by any of the machines infected with malware or bantenets can make false claims against servers, similar to those seen in DDoS attacks. In the advanced type of these attacks, it is observed that the botnets do not focus against the server, but act against the links connected to the server, as can be seen in CrossFire attacks. A good way to detect botnets is to use security layers on the bottom and bottom layers of the Internet of Things. These methods try to analyze the data in the sensor layer of the IoT network, and if this data and traffic have a pattern of bnetnet attacks, then they are filtered.

1.3. Motivation and importance of the study

DDoS attacks by malicious botnets and nodes on the IoT network are among the most malicious attacks on the network and can disable IoT application servers. The importance of identifying these attacks is that a machine or node infects a large number of machines on the network with malware and then launches attacks against network services, challenging network performance to provide customer service. Therefore, it is important to create a security layer on the surface of SDN switches because it does not allow traffic to be sent from the bottom layers to the servers in the top layer or cloud.

1.4. Objective of Research

This research has several objectives, some of which are listed below:

1. Identify attacks in smart cities
2. Use SDN architecture and counter network intrusion at lower layers
3. Sharing the IP address of botnet machines in SDN switches and not allowing these machines to access the network cloud computing layer
4. Increase the security of the smart city network by integrating deep learning methods

1.5. Organization of the Thesis

This thesis includes five chapters. At first chapter the introduction and the proposed study is presented. Literature review and the work studies that worked in research studies explained in chapter two. The methodology and material of study is presented in chapter three. Chapter four contains the results and discussion. Finally with conclusion and future work this study will be finalize.

2. LITERATURE REVIEW

This chapter explains in detail proposed method and gives an overview of prediction methods.

2.1. Background

This section reviews and analyzes concepts related to the Network attack detection article. In the beginning, the algorithm of whale optimization, Harris Hawks Optimization, and jellyfish optimization are introduced, and then relevant research in the intrusion detection field in IoT and finally, software-based networks employing techniques like deep learning and machine learning are compared.

2.2. Literature Review

According to the studies, the detection of attacks on the network is important at least in terms of reducing the harm of attacks and its effects, and this research is in this direction and a number of studies in this field are reviewed below:

In the study (Putra et al., 2020), in 2020, they introduced a scalable and reliable decentralized participatory intrusion detection system for the IoT network. An intrusion detection system can issue alerts to users about attacks by deploying a tracker that continuously monitors network traffic. In most studies, an attempt has been made to have a network system play the role of a network intrusion detection system, but in this study, a group approach has been used to detect network intrusion. In the proposed method, several systems in the network and their participation are used to detect network intrusion. However, creating distributed and effective intrusion detection systems, especially for the Internet of Things, poses many challenges. Recent trends and advances in blockchain technology that ensure the distribution of trust and secure immutable storage can be well used in the design of group and distributed intrusion detection systems. Their studies show that the use of China Blockchain technology to detect intrusion can increase the effectiveness of this technology in detecting common attacks on IoT networks.

In the study (Eskandari et al., 2020), in 2020, they introduced an intrusion detection system based on intelligent anomalies for devices connected to the Internet of Things. Protecting cyber threats is one of the most challenging areas of information technology research today, and the number of Internet-connected devices that can transmit personal data over the Internet is growing dramatically. IoT protection is very important and vital because there are usually several IoT-based data sources in different applications in this network, and attacking any of these objects can make the use of the Internet network challenging. In this research, an intelligent intrusion detection system called Pasban has been presented and introduced to protect the IoT network. An important feature of their proposed solution is that it can be installed directly on the very cheap gateways of the Internet of Things. Their proposed intrusion detection system is capable of various types of malicious traffic, including port scanning, password guessing attacks and Identify flood attacks on the Internet of Things at high speed and with very low false positive rates and satisfactory accuracy.

In the study (Z. Liu et al., 2020), in 2020, they introduced the detection of anomalies in the penetration of the Internet of Things using machine learning. Increasing the security of IoT networks is one of the most important issues facing the IT community. With the development and deployment of large-scale IoT devices, the ability to communicate securely and without compromising on the performance of these devices has become a challenging issue. There are challenges in IoT network security because most IoT devices have limited power, in other words, their limited computing power makes it impossible to support strong security systems. Due to the weakness of the objects connected to the IoT network, it is difficult to encrypt and authenticate to prevent malicious cyber-attacks on this type of network. An intrusion detection system logically as a leading security solution can be of great help in solving this challenge. Network intrusion detection can be based on the non-intelligent mechanism of blacklisting or it can be based on anomalous learning methods. Machine learning and anomaly detection mechanisms play an essential role in protecting networks against various malicious activities. In this study, they applied different machine learning algorithms to effectively identify anomalies in the Internet penetration data network, and their results show that machine learning methods are highly accurate in penetration detection.

In research (K. Liu et al., 2020) and in 2020, a review of Android malware detection approaches based on machine learning is presented. Android apps are evolving rapidly in the

mobile ecosystem, but Android malware is also emerging endlessly, and many researchers have explored the issue of Android malware detection and theories and methods from different perspectives. Have provided. Existing research shows that machine learning is an effective and promising way to detect Android malware. This article provides a comprehensive overview of machine learning approaches based on machine learning. They briefly introduced some areas related to Android applications, including Android system architecture, security mechanisms, and Android malware classification. This review will help academics get a complete picture of machine-based Android malware detection. They can then serve as a basis for future researchers to start new work and help guide research in this area.

In the study (Chakravarty, 2020) and in 2020, presented feature selection and evaluation of licensed Android malware detection. Android malware is a pervasive threat to the security of mobile users' information. Android users often download apps from unauthorized and unreliable sources. Such programs may require multiple permissions from the user, and due to lack of awareness, the user can grant the necessary permissions. Android licenses are one of the most important sources of malware infection. By analyzing the permissions, it is possible to classify malware and malware in the database with the help of machine learning tools. There are a total of 330 licenses in Android apps. However, not all of them may contribute to the classification. In this paper, the proposed system examines the identification of the most effective permissions using feature reduction. In this research, decision tree, random committee, perceptron multilayer, sequential minimum optimization and random classifiers are used to evaluate the selected features. Experimental results show that five licenses can be almost complete, thus optimizing the malware detection system.

In the study (Haddadpajouh et al., 2020) and in 2020, they introduced a meta-heuristic and multi-kernel feature selection approach to identify the threat of IoT malware in the sensor layer and smart objects. IoT devices are increasingly targeted by malware due to their presence in a wide range of applications, including at home and in corporate environments. In this paper, they propose a method for detecting malware on IoT devices using the Gray Wolf optimization method and a multi-core support vector machine. This model is trained with IoT sample malware codes. The training data set consists of 271 benign samples and 281 malicious samples of Cortex A9 and is evaluated using cross-validation method. They validated the proposed model in terms of its ability to detect IoT malware that had not been

seen before. Their proposed model, compared to the previous model of deep neural network, which requires more than 80 seconds to train the same data, requires only 20 seconds to train. In general, the proposed multi-core backup vector machine method performs better in terms of accuracy than deep network and IoT malware detection techniques based on fuzzy methods, while reducing computational cost and training time significantly reduced.

In the study (Pundir et al., 2021) and in 2021, they presented a protocol for detecting malware attacks in the industrial multimedia environment on the Internet of Things using a machine learning approach. One of the technologies that is developing the fastest is the Internet of Things. Almost all daily uses are now connected to the Internet thanks to the development of vast and quick mobile networks. The collection and analysis of many types of multimedia data takes place in an industrial multimedia environment that can link to the Internet of Things. In order to monitor and regulate industrial processes, a variety of intelligent equipment, including drones, robots, intelligent controllers, and intelligent monitoring systems, produce this multimedia data. Multimedia data, which can be categorized as big data, are produced in vast quantities today. The majority of uses for this data are for various commercial requirements. Because every smart equipment is connected to the Internet and may be controlled there, there are serious communication risks in an IoT-enabled industrial multimedia environment. It is susceptible to numerous forms of attacks, including malware injection, forging, information leakage, modifying sensitive information, and retransmission. Therefore, it's critical to guard against various potential threats from entering such an environment. Researchers' attention has recently been drawn to attacks carried out using botnets. Such assaults interfere with the communication of IoT-enabled industrial multimedia environments. Additionally, attackers have the ability to modify the functionalities of smart devices and control them remotely. Therefore, a robust mechanism is required to identify malware attacks in such a setting. In this research, they provide a machine learning method known as MADP-IIME for malware detection in an industrial multimedia environment that is IoT enabled. To accurately identify malware attacks, the suggested solution employs four different Bayesian machine learning techniques: logistic regression, artificial neural networks, and random forests. Additionally, the suggested approach outperforms competing comparable systems and yields a diagnosis with a 0.5 false positive rate. The security research that was conducted also shows how adaptable the suggested approach is to various malware threats.

They introduced deep learning based on short-term memory and fast learning techniques to identify malware in IoT applications in the study (Khowaja & Khuwaja, 2021) and in 2021. In the Internet of Things context, electronic edge devices or smart objects are frequently utilized as bridges between the device and the service layer. These systems are totally open to malware attacks. Existing research on the creation of complex learning algorithms or thorough structures for the classification of malware is labeled, presuming there are enough examples. One of the primary subjects in the training of any classification system in the actual world is getting labeled data. Researchers can now detect malware using active learning techniques thanks to recent advancements. In this study, rapid learning features are incorporated into an active learning framework that enables the network to request or forecast a sample-related tag during training. To categorize malware apps while employing a few samples of tags, they employ the self-encoder network and the LSTM deep learning method with active value function. When requesting or predicting tags, this network depends on its uncertainty. The proposed method can achieve better accuracy than the supervised learning strategy while using multiple label requests, according to experimental data. The outcomes also demonstrate the taught network's resistance to hostile assaults, demonstrating the resilience of the suggested approach. Through the choice of awards and the use of decision-level fusion algorithms to improve classification performance, this study investigates the trade-off between classification accuracy and a number of label requests. As concepts for the suggested strategy, they also provided a hypothetical framework.

In the study (Arul, 2021) and in 2021, presented a polynomial method of least squares deep nonlinear regression for malware detection on the Internet of Things and smart devices on the network. Innovative techniques need to be developed to reduce the risk of malware spreading on the IoT network. In this paper, a deep nonlinear regression method of least polynomial squares is proposed and through this prediction model, an attempt is made to identify objects and devices infected with malware. Through two popular botnets, Mirai and BASHLITE, they introduced nine smart tools and tools to improve malware detection performance using the deep learning method. The identification and analysis of malware in industrial applications of the cyber-physical system, which offers numerous services utilizing the IoT model, has raised concerns due to the rapid development of IoT technology. Many cutting-edge machine learning techniques, like in-depth learning, are currently common in malware analysis and detection research, and progress has already been made in this area. There are, however, some issues as well. For instance, certain methods are

insufficiently reliable because of the noise and radiation in the current malware database. Therefore, there is still room for improvement in the accuracy of malware classification. In order to do this, we suggest a novel technique that fuses the displacement method with a deep learning model. In their proposed method for identifying and analyzing malware, the Success Index is used as an effective measure of similarity in dealing with complex datasets with noise and learning with convolutional neural networks.

In the study (Luo et al., 2021) and in 2021, they presented an efficient combined framework for the detection of deep neural network-based malware in a fifth generation network environment. Today, while Android smartphones are widely used in Najm generation networks, personal application operating systems are facing a rapid increase in the display of applications to launch the market. This paper provides an effective framework for identifying deep neural network-based malware called DLAMDs that can encounter large-scale examples. An efficient diagnostic framework is designed, which combines the rapid detection stage with in-depth diagnostics. A convolutional neural network that automatically extracts the hidden pattern inside the features and a random forest with a favorable effect are chosen to pick the importance in order to provide a feature subset that can further differentiate essential features. , is chosen in order to choose a feature. This experiment examines the high efficacy of the suggested strategy using actual data from popular malware sets and operating systems for downloading personal applications. The findings indicate that this strategy can achieve an accuracy index for malware detection of 95.69%.

One of the most widely used and well-known technologies in the twenty-first century, the Internet of Things (IoT) has experienced an explosion in popularity in recent years. Anyone may connect everyday items like baby screens, car regulators, kitchenette apps, and autos to the Internet by employing embedded technologies. As a result, there may be cohesive communication between individuals, processes, and objects [29-(Gupta & Quamara, 2020).

Low-cost computers, the cloud, big data, analytics, and smartphone expertise allow physical items to collect and store data with the least amount of human intervention. In this highly interconnected environment, digital systems have the power to enhance, filter, and switch any communication between connected objects. With the help of the digital world, the physical world interacts and works together.

A variety of limitations apply to the investigation of the transmitted data. Regular domestic internet service providers do not have the time or technical know-how to monitor their network activity, therefore if the data is encrypted, it will not be clever to recite the data that has been sent to them. In a normal household, more than 10 smart home appliances that connect to the Internet can be used simultaneously, and this number is rising (Stolfo et al., 2000).

A revolutionary uprising against the internet, the Internet of Things. It decides to become knowledgeable and to make them known. Due to their capacity to connect facts about oneself, they may either make decisions for themselves or allow others to do so. The information that has been gathered by others may be altered by them. Things or mechanisms may be highly developed facilities. This change signals the arrival of cloud computing and Internet broadcast capabilities based on IPv6 and its virtually infinite processing power (Alkasassbeh et al., 2016)- (Schull et al., 1981).

Companies should start educating themselves about and utilizing the opportunities provided by the IoT. It is possible to enhance operations, develop new revenue sources, and regain customer service by gathering and analyzing data from dispersed linked devices (H. Liu et al., 2011). Additionally, because of increased network connections and poor project design, the Internet of Things exposes governments to new security gaps. Attackers with more experience can now switch to attacking other systems (Khammassi & Krichen, 2017).

In order to detect IDS in IoTs, this thesis aims to provide a novel framework and the best data mining techniques. The suggested framework will be tested using widely used datasets in this field, and the outcomes will be contrasted with the most effective methods that have already been published. Deep learning applications in other fields of study, including as picture identification, disease detection, and data categorization, have so far produced results that are acceptable. For identifying IDS attacks in gas and oil IoTs, a deep learning technique has been refined. Data mining techniques are not very accurate in complex scenarios. The study's main advance, however, is enhanced data mining, which enhances technique accuracy.

Through embedded devices, actuators on machines, and more physical sensors, Internet of Things (IoT), smartphone-connected systems, and schemes all affect data security. This

conference will introduce GSMA "Related Life," a new facility that improves consumer quality of life and company production. The Internet of Things is predicted to grow fast in the coming years. Numerous technologies make the use of IoT more difficult (Dhieb et al., 2020).

The basic idea of an intelligent device network was investigated in 1982, when a customized Coca-Cola chocolate appliance at Carnegie Mellon University became the first ARPANET-connected device (Machine, 2014), capable of describing its stock and determining the temperature of recently piled beverages (Palermo, 2014). The 1991 essay "The Computer of the Twenty-First Century" by Mark Weiser as well as academic settings like UbiComp and PerCom informed the current concept of the Internet of Things (Friedemann & Floerkemeier, 2010).

Utilizing these platforms enables municipal residents, local governments, and other stakeholders to collaborate on issues like parking, traffic, and lighting while making better use of their time and other resources. They also facilitate improved waste management. Applications exist for producing and sharing predicted traffic densities for the present or the future. You can learn about available parking places in the area you're visiting, along with their occupancies and parking costs, by using mobile parking apps. They could provide information on waste management practices like tracking trash container fill rates, improving garbage truck collection processes, moving container points, and changing container sizes (Aktemur et al., 2020).

Applications for monitoring environmental factors like air pollution, precipitation, dam fullness, and forest fires are being created in order to take prompt action as necessary. Smart home applications are being developed to give consumers the ability to remotely control and check on the state of the services their homes provide in a variety of sectors, including lighting, heating, ventilation, entertainment, and security, among others. It is being developed in order to support procurement processes including the monitoring of products whose stock is running low and whose expiration date is drawing near, as well as the restocking of products if necessary. It is being created to keep an eye on things like temperature and the presence of dangerous gases in places where there are animals, among other things. In The goal of IoT efforts is to deliver capabilities or value to the organization's business layer. A vast volume of sensor data (data) from the physical environment is

collected, and analysis systems are developed that include the necessary analytical architecture and mechanisms for communicating the information from the analyzed data to the business layer. IOT include tasks like selecting the sensors to use for data collection, choosing the door to use for data transmission to focus personnel, information communication standards, and decision system setup (Bhushan & Sahoo, 2020).

The sensors that will be incorporated into the systems could be fundamental devices like RFID tags, straightforward sensors like a thermometer circuit that shows the outside temperature at specific times, or decision-making and operational systems like a camera that focuses on motion. Preprocessing, storage, and archiving are optional but all of these steps—starting with data production and continuing through aggregation, transmission, and filtering—are necessary. In spite of the fact that data is produced (requested) and consumed (analyzed), querying and analysis are the times when these activities occur. Data gathering, aggregation, filtering, and early processing are among the online, communication-intensive processes (Carrez, 2013).

2.3. Whale Optimization Algorithm (WOA)

Humpback whales are a group of types of whales in which social behaviors can be seen in a significant way. These creatures have spindle cells similar to humans and are capable of making decisions and learning. These creatures have interesting group and social behaviors in hunting clusters of fish and they try to hunt in groups for this purpose they use the Hunt Bubble mechanism. Hunt Bubble in humpback whales is actually a group behavior. The WOA is a group algorithm based on the mass hunting of whales and in this algorithm, the behavior of whales is used to find the optimal point, which is the aggregation of fish, to solve optimization problems. In order to hunt a cluster of fish or guide them to the optimal point, whales use two methods of hunting: upward-spirals and double-loops to surround the cluster of fish in order to place the cluster at an optimal point and hunt them (Mirjalili & Lewis, 2016). This algorithm uses upward spirals and double loops in order to seek out the problem space and assumption of the algorithm is that every problem solution is a whale that looks for the optimal solution or food with the help of other whales. Figure 2.1 shows the rotating and spiral movement of the whales to approach the mass of fish or the optimal solution and search around the optimal solution (Su et al., 2022).

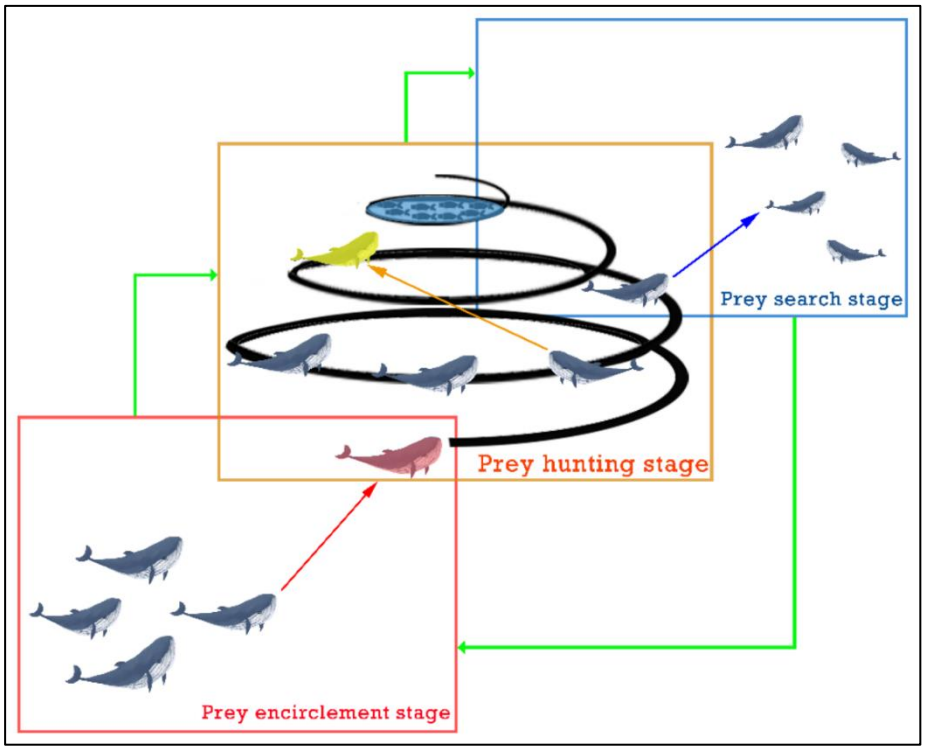


Figure 2.1. Rotational and spiral movements of the whales around the prey (Su et al., 2022)

Due to the correct search of the problem space, WOA finds the best solution with great accuracy in both rotary and spiral movements and it is more accurate than the Genetic Algorithm, Evolutionary Difference Algorithm and Particle Swarm Optimization Algorithm.

2.4. Harris Hawks Optimization (HHO) Algorithm

The current position is actually considered as prey in the Harris Hawks Optimization (HHO) Algorithm, and each hawk is also a problem solver that searches around the optimal solution to find more optimal answers. Creating diversity of hunting behavior in hawks and not limiting the hawk to movements such as diving, increases the accuracy of this method in finding the optimal solution. In the Harris Hawks optimization algorithm, there is a set of behaviors that can be used to solve the problem and guide the population. In the HHO algorithm, each hawk can slowly place itself around the prey and move towards the prey over time, and in this case, the hawk dives from a high altitude and is placed in an area far from the prey and by reducing its height it moves slowly towards the prey. According to the HHO algorithm, the Soft besiege encirclement behavior involves hawks slowly moving toward the prey and scanning its surroundings. In figure 2.2, a set of complex behaviors of

hawks to search the problem space is displayed. In the HHO Algorithm, each solution moves towards the prey with random movements, slow encirclement and fast diving.

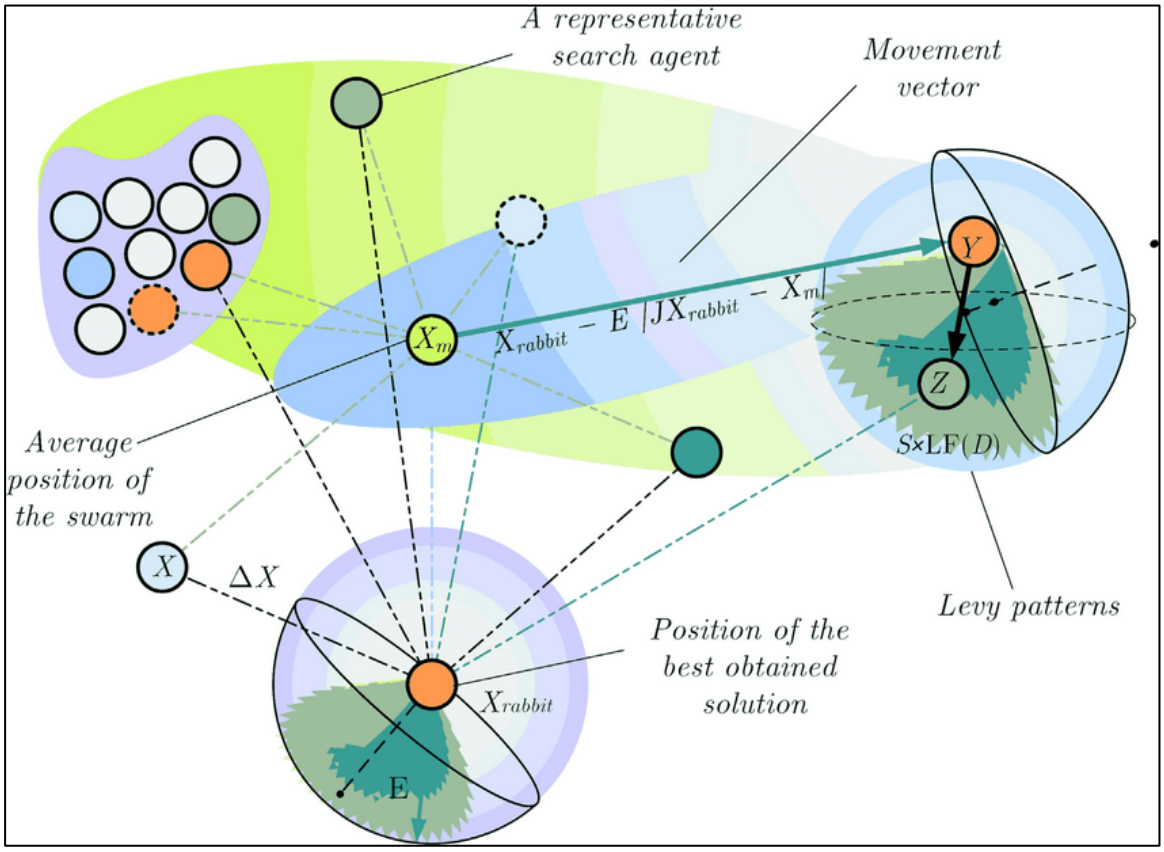


Figure 2.2. Movements of the hawk to search around the prey (Song et al., 2021)

Studies and trials reveal that their suggested algorithm outperforms the genetic algorithm, particle swarm optimization (PSO), firefly algorithm, bat algorithm, biogeography-based optimization (BBO), cuckoo search, and differential evolution in terms of accuracy.

2.5. Jellyfish Search Optimizer (JSO) Algorithm

The JSO algorithm is a meta-heuristic algorithm presented in 2021 and modeled on the swimming behavior of jellyfish. In this algorithm, each problem solution is a jellyfish that looks for food or an optimal solution. Jellyfish or problem solvers have several types of moves to find the optimal solution. Their first movement is based on sea water waves and their second movement is based on group behavior in groups. The JSO algorithm, as shown in Figure 2.3, is modeled from the group behavior of jellyfish and the motion change of jellyfish with water waves (Chou & Truong, 2020).

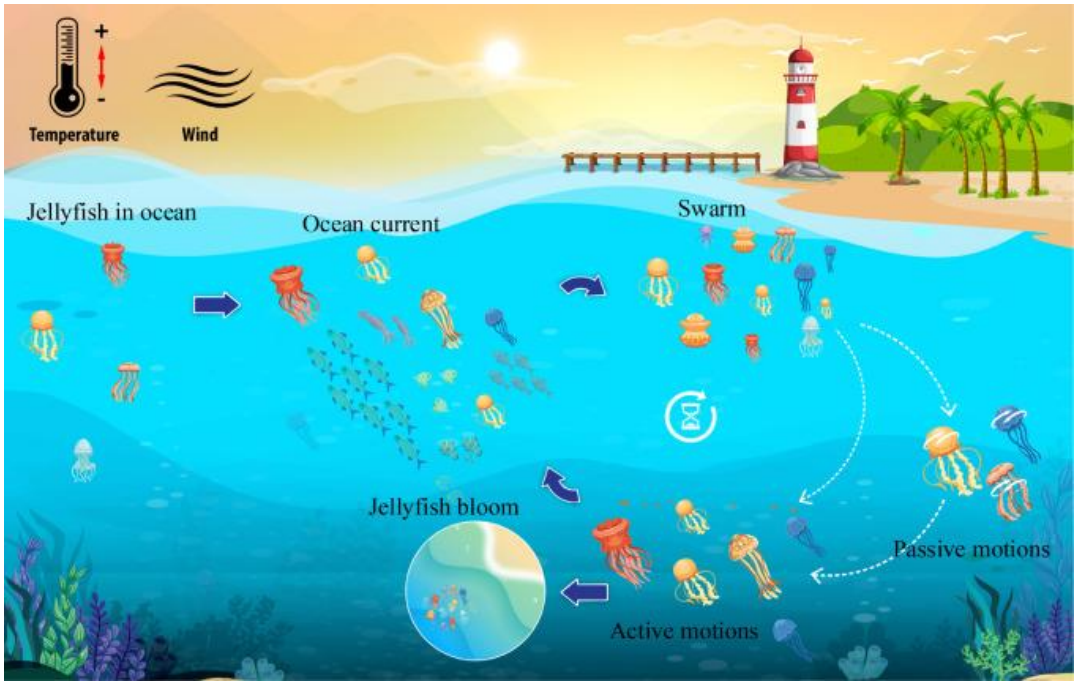


Figure 2.3. Jellyfish behaviors in JSO algorithm modeling (Chou & Truong, 2020)

This algorithm has types of local and global search to find the optimal solution, and on the other hand, it has been proven to be more accurate than genetic and particle algorithms.

2.6. SDN Network

Software-defined networks or SDN have created a great revolution in computer networks. The software-oriented network can be considered a new network architecture for the logical separation of the control level (decision-making) and data in the network. Software-based network architecture was first proposed by Stanford University and then developed by other companies. In the past, Ethernet switches were used to route packets in the network, and these hardware's had a processing unit to decide on sending data, and they are called control plane. For configuration in this type of networks, managers needed to spend a lot of time. To solve these problems, the software-based network has been introduced and is supported by various organizations, one of which is Open Networking Foundation (ONF). ONF stands for an organization that aims to promote and popularize SDN using open-source standards to implement software-oriented networks. Software-oriented network or SDN, like traditional networks, has a set of network equipment such as switches, routers and middlebox appliances. In this type of networks, the control plane is referred to the intelligence of the

network, and the data plane is also defined as the executive arm of this intelligence. SDN network architecture is shown in Figure 2.4 (Wu et al., 2020).

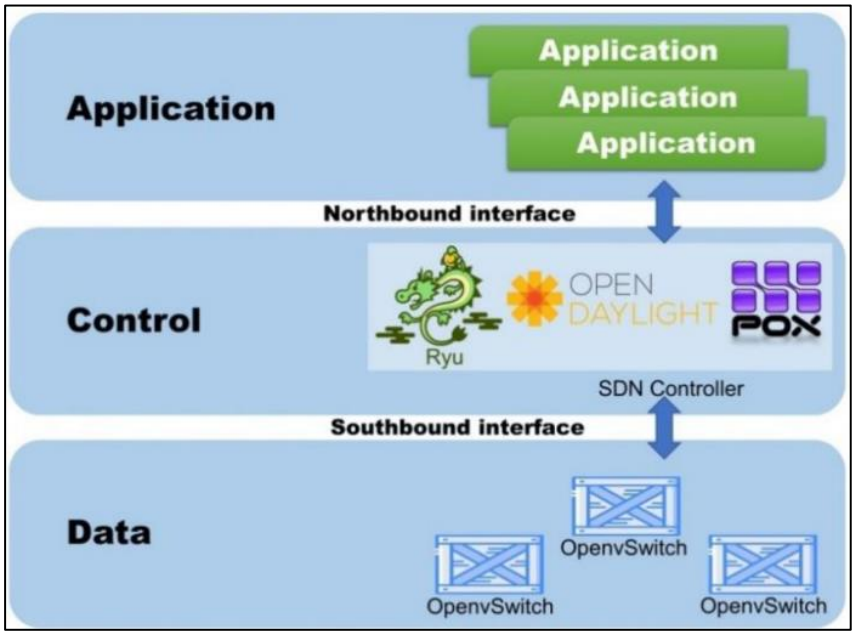


Figure 2.4. Software-defined network layers (Wu et al., 2020)

The control system in the software-defined network includes the network operating system and applications. There are three sections: internet, control unit and application section. It has a three-layer architecture for sending and directing data. The lowest level of this architecture is related to data, the middle layer is related to data control and finally, the application section is located in the upper layer. It can be seen that in the application layer and the software-oriented network layer, there is a two-way communication for data flow control, the data is located in the data layer and the control of sending data is defined in the software-oriented network controller layer.

2.7. Related Works

In the study (Kranthi et al., 2022), they presented a case study about network intrusion detection systems in software-based network architecture using the concept of machine learning. The results of their research show that deep learning methods as a subset of machine learning methods have a greater ability to detect attacks and use them in network intrusion detection systems. In the study (Prabakaran et al., 2022), 2022, they presented a method to predict the pattern of attacks in the IoT network and through machine learning in

the software-oriented network. In this study, machine learning algorithms are trained using traditional network threat intelligence data to identify potential malicious links and potential attack targets. The results of the experiments show that the Bayesian network algorithm has an accuracy of 92.87% in detecting attacks and is more accurate than the decision tree methods in detecting network attacks. In the study (K.-H. Le et al., 2022), 2022, they proposed and introduced an intelligent intrusion detection system against cyber threats in the IoT. The core of their proposed intrusion detection system is a lightweight convolutional neural network model to classify multiple cyber threats. The evaluations showed that their proposed method detects 9 types of cyber-attacks (such as back doors, shell code, and worms) with an average accuracy of 97.22%. In the study (Gu et al., 2022), 2022, they presented and introduced a new intrusion detection system for industrial control systems. In this study, they presented a reconstructed convolutional neural network to design a network intrusion detection system. Evaluations show that the accuracy of their proposed method for detecting network intrusion is about 97%. In the study (Thirimanne et al., 2022), 2022, researchers proposed a real-time intrusion detection system that uses deep neural networks. A deep neural network that has been trained using 28 features from the NSL-KDD dataset makes up the suggested system. The suggested method's respective accuracy, sensitivity, and specificity indicators for identifying network penetration are 81%, 96%, and 70%. The important advantage of this intrusion detection system is that it is able to identify modern network intrusions. In the study (Imran et al., 2022), 2022, they proposed a network intrusion detection system using cuckoo search optimization. Experiments showed that their proposed method of fuzzy clustering artificial neural network, intrusion detection with an artificial bee colony has more accuracy in intrusion detection. In the study (Hosseini & Sardo, 2022), 2022, they modeled and proposed a network penetration detection based on the deep learning method in the IoT. The suggested approach uses the spider monkey optimization feature selection algorithm to first identify the most crucial properties, and then it suggests a Siamese neural network model to further categorize the data. The accuracy of the proposed model using random forest classification is 94.69%. In the study (Alzahrani & Bamhdi, 2022), 2022, they introduced and presented a hybrid deep learning model to identify botnet attacks in IoT environments. The proposed system of attack detection results can have a high accuracy of about 88%. Their evaluation shows that their security and intelligence system using an advanced deep learning model was successful in detecting botnet attacks infecting camera devices connected to IoT applications. In the study (Dora & Lakshmi, 2022), 2022, they presented a security and intrusion detection system to deal with

DDoS attacks by selecting the optimal feature with feature learning by convolutional learning network and deep learning based on LSTM network. The proposed intrusion detection system is tested on a set of benchmark data sets and the results were compared with traditional models, which shows that this method is more accurate than intrusion detection systems such as machine learning.

The examination of related studies and works reveals that deep learning, feature selection and machine learning techniques have been employed in the majority of studies to detect SDN network penetration. Deep learning methods to detect attacks require a lot of training and resources and have an automatic feature selection mechanism with no intelligent feature selection phase. Machine learning methods in combination with feature selection methods are effective and require less time for training, but their accuracy depends on the feature selection method. The review of studies shows that a feature selection method has been used in most of the studies and if the feature selection algorithm selects non-essential and essential features, then the error of the intrusion detection system will increase. Another challenge of the reviewed studies is that none of them used blacklist and cooperation between SDN switches. Considering the gaps in the proposed method for detecting attacks in SDN, a feature selection system with majority voting mechanism and blacklist exchange between SDN switches is presented.

3. MATERIAL AND METHODS

Smart monitoring with SDN switches can increase the security of the network and its users. A common way to increase network security is to use network intrusion detection systems. Intelligent monitoring with SDN switches to increase the security of network users is actually a classification problem that needs to classify network traffic in different categories. In most of the studies, all dimensions and characteristics of network traffic have been used to classify network traffic, but in this research, first an SDN switch discovers the important characteristics of network traffic by majority voting in three methods, HHO, WOA, and JSO, to analyze network traffic. This method reduces the volume of network traffic and then the reduced traffic (by selecting the feature) is considered as the input of machine learning methods. In the research, each SDN switch uses a machine learning method to analyze the network traffic and if it detects an anomaly in the network traffic, then it sends the originating IP of the traffic sender to the firewall to be blocked. The firewall also sends the list of IPs reported by SND switches to other SDN switches and controls so that they prevent the traffic of these IPs from being transmitted. In the proposed method, machine learning techniques such as artificial neural networks are used to analyze and evaluate network traffic.

3.1. The Proposed Framework

In Figure 3.1, the proposed framework for detecting network attacks with SDN architecture is shown and according to the figure, it has the following steps:

- NSL-KDD data set traffic is considered as the input of SDN switches.
- Traffic is pre-processed and normalized.
- Traffic is divided into two categories: training and test.
- A feature vector is considered as a member of WOA, HHO, JSO algorithm.
- An initial population of feature vectors is created in WOA, HHO, JSO algorithms.
- WOA, HHO, JSO algorithms are implemented on the feature vectors and each group intelligence algorithm updates its related feature vectors.
- The most optimal feature vector is calculated by WOA, HHO, JSO.

- The final feature vector is selected by majority voting among the most optimal feature vector calculated by three methods WOA, HHO, JSO.
- The optimal feature vector is sent by the controller to the SDN switches.
- Each SDN switch receives the optimal feature vector to reduce traffic dimensions and trains a multi-layer neural network.
- The trained neural network model with the optimal feature vector is used in each SDN switch to evaluate the network traffic.
- If a traffic is an attack type, the IP node associated with the attacking traffic is added to the blacklist and the blacklists are exchanged between SDN switches.
- Finally, the suggested approach is assessed using metrics like accuracy and error.

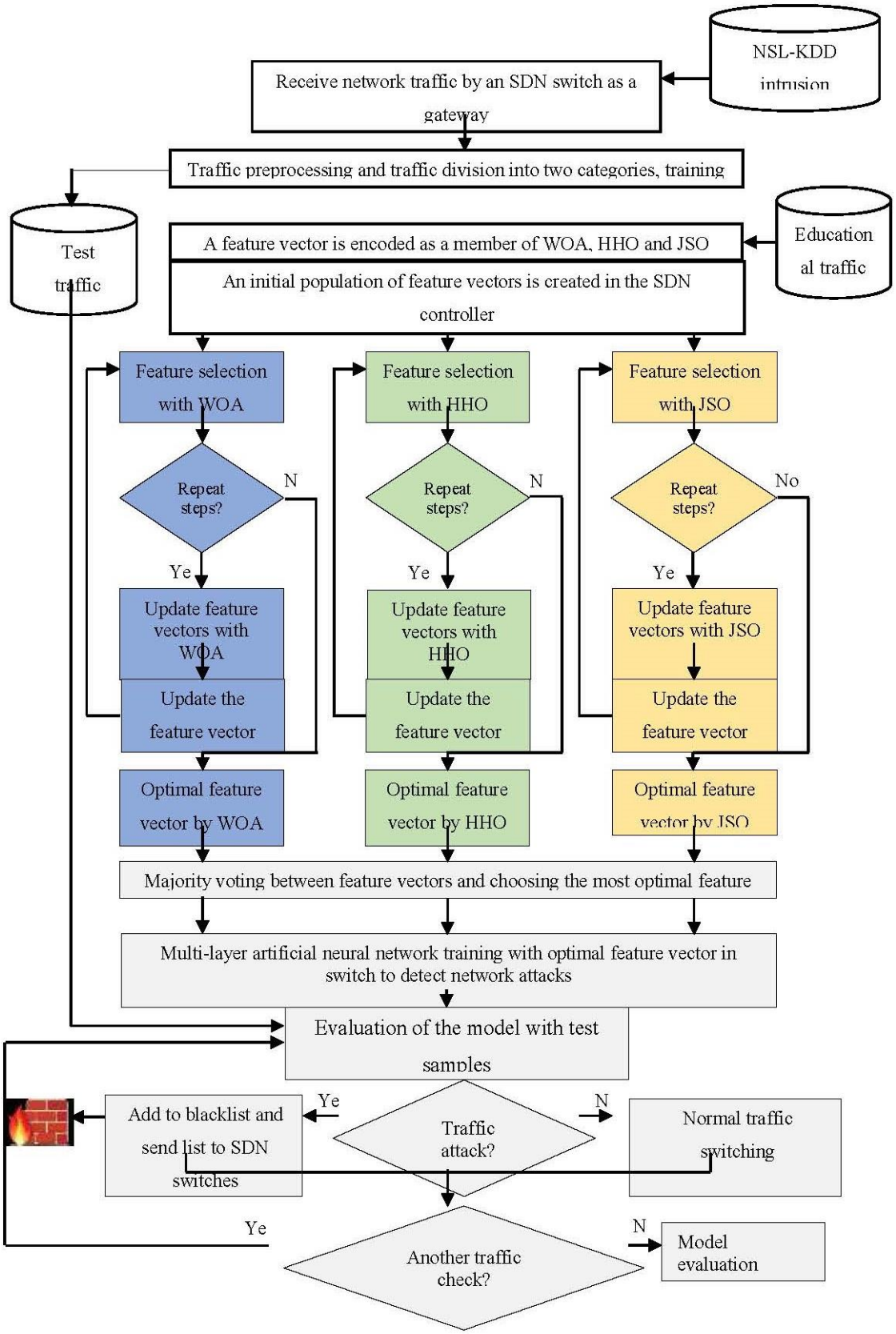


Figure 3.1. Conceptual model of the proposed method in detecting attacks in SDN network

3.2. K-NN Algorithm

In all methods the k-nn used to classification of the data inside of the algorithm. K-Nearest Neighbor, one of the simplest machine learning algorithms, employs the supervised learning approach. Based on the presumption that the new instance and the data are comparable to the examples that are already available, the K-NN algorithm places the new case in the category that is most similar to the categories that are now available. The K-NN method categorizes fresh input based on similarity while saving all previously collected data. This means that as new data is produced, it may be promptly categorized using the K-NN algorithm into an appropriate category. Although the K-NN technique can be applied to both classification and regression problems, classification challenges are where it is most frequently used. Since the K-NN approach is non-parametric, no presumptions are made on the underlying data. This algorithm, often known as a lazy learner because it does not immediately learn from the training set, keeps the dataset and performs an action on it when the time comes to categorize. During the training phase, the KNN algorithm simply stores the dataset and then sorts incoming data into categories that are similar to those in the previously stored dataset.

3.3. Network Traffic Preprocessing

The network intrusion detection dataset needs pre-processing such as normalization so that all samples have numerical values and limited in a normal interval such as [a,b]. To normalize the data set, the linear relationship in equation (1) can be used:

$$N' = \frac{N - min}{max - min}(b - a) + a \tag{3.1}$$

In normalization, $N'$is equal to the normalized value of a network traffic feature, N is equal to the non-normalized value of a network traffic feature and min, max are the lower and upper limits for normalization, respectively. Normalization causes all the used features to be changed only in a certain range and learning is done more accurately.

3.4. Objective Function

In the proposed intrusion detection system, using WOA, HHO and JSO algorithms, the optimal feature vector can be selected by majority voting and this feature vector can be used for training and learning artificial neural network in SDN switches. A feature vector that can better describe network traffic should have at least the following two important features:

- Minimal error in normal and abnormal traffic classification.
- The number of selected features should be as minimal as possible.

To describe the network intrusion detection error, objective function number one in relation (2) can be used:

$$F_1(X(t)) = \frac{1}{n}\left(\sum_{i=1}^{n}(\tilde{y}_i - y_i)^2\right) \tag{3.2}$$

In this objective function, it is assumed that there are two normal and abnormal classes in the network traffic, which are indicated by zero and one, respectively, and the target problem is a classification problem and two classes. In the proposed method, the estimated value for the class of a sample like i is equal to $\tilde{y}_i$ and the actual class of the sample for a given data is equal to y_i. The second objective function is also the average number of selected features to all the features used, and as in relation (3), this objective function is determined:

$$F_2(X(t)) = \frac{\|X(t)\|}{m} \tag{3.3}$$

In this regard, $\|X(t)\|$ is the number of features selected in the feature vector $X(t)$and m is the number of main features in the data set, and if the KDD data set is used, then this parameter will be equal to 41 . These two goals can be considered in the form of a single goal like equation (4), and the optimal feature vector to reduce network traffic dimensions is the feature vector which minimizes this function (Emine & Ülker, 2020):

$$F = \alpha . F_1\big(X(t)\big) + \beta . F_2(X(t)) \tag{3.4}$$

Alpha and beta are two weighted values in this objective function between 0 and 1, whose total is 1.

3.5. Feature Selection With WOA Algorithm

The proposed method for feature selection using Whale Optimization Algorithm is shown in Figure 3.2, and its steps are as follows:

- Network traffic is considered as SDN input.
- Network traffic is pre-processed and normalized.
- Coding a feature vector as a member of the algorithm
- Creating an initial population of feature vectors as the initial population of the WOA template
- Evaluation of feature vectors with the objective function
- Selection of the most optimal feature vector in the first step
- Feature vectors are updated based on random, spiral and rotational behaviors.
- Feature vectors with binarized S and V functions
- The feature vectors are evaluated with the objective function.
- The most optimal feature vector after updating the determined feature vectors
- WOA algorithm parameters such as A are updated in each iteration.
- Algorithm repetition counter increases by one unit in each repetition.
- The optimal feature vectors in the final iteration participate in majority voting using HHO and JSO methods.
- With Whale's approach, the suggested feature selection method is assessed.

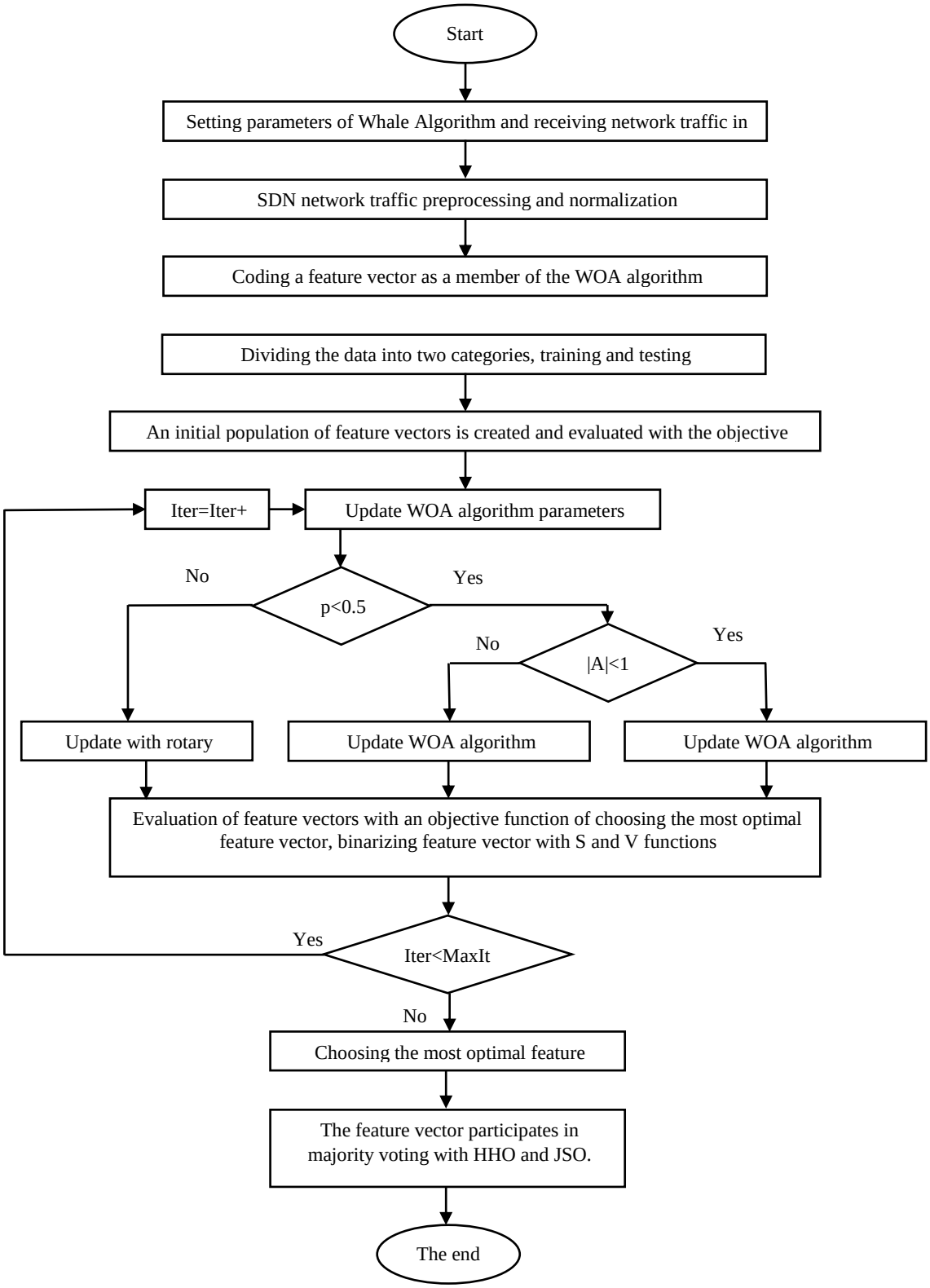


Figure 3.2. Feature selection flowchart with Whale's algorithm in suggested approach

In the proposed feature selection approach, an optimal solution or an optimal feature vector in iteration t is represented by$\vec{X}^*(t)$. In the proposed method, if Whale's algorithm is used

for feature selection, then a whale is a feature vector that can perform circular and random movements to be updated. To update the feature vectors with rotational movement, first a whale calculates its distance from the optimal whale as in relation (5) and then as in relation (6), it performs rotational movement in line with this whale:

$$\vec{D} = \left|\vec{C}.\vec{X}^{*}(t) - \vec{X}(t)\right| \tag{3.5}$$

$$\vec{X}(t+1) = \vec{X}^{*}(t) - \vec{A}\vec{D} \tag{3.6}$$

In these relationships, $\vec{D}$ is the distance vector of the position of a feature vector or a $\vec{X}(t)$ whale from the optimal position or the optimal feature vector such as $\vec{X}^{*}(t)$ in the t iteration of the proposed algorithm. In the proposed method, relation (7) and (8) are used to select the feature and to calculate the parameter $\vec{C}$ and $\vec{A}$:

$$\vec{A} = 2\vec{a}.\vec{r} - \vec{a} \tag{3.7}$$

$$\vec{C} = 2\vec{r} \tag{3.8}$$

In these relationships, $\vec{a}$ is one between two and zero, and over time and according to repetition, it should decrease from two to zero. In this relation, $\vec{r}$ is also a random vector with a uniform distribution in the interval [0,1]. To update feature vectors with spiral movements, equation (9) is used:

$$\vec{X}(t+1) = \overrightarrow{D'} + e^{bl}.\cos(2\pi l) + \vec{X}^{*}(t) \tag{3.9}$$

In the relationship mentioned above, $\overrightarrow{D'}$ is the distance between a feature vector's or whale's current location and the whale's or feature vector's optimal position. The parameter l is a uniform random value in the range [-1,+1] and the parameter b is a constant value. In a spiral movement, equation (10) is used to calculate the vector $\overrightarrow{D'}$:

$$\overrightarrow{D'} = \left|\vec{X}^{*}(t) - \vec{X}(t)\right| \tag{3.10}$$

To perform spiral or rotational movements on each feature vector or a whale in each repetition in equal proportion and as in relation (11), these two equations are used to update the feature vectors:

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A}\vec{D} & if\ \ p < 0.5 \\ \overrightarrow{D'} + e^{bl}.\cos(2\pi l) + \vec{X}^*(t) & if\ \ p \geq 0.5 \end{cases} \quad (3.11)$$

In this regard, p is an accidental value between 0 and 1, which is generated for each feature vector or a whale. If the random number is greater than 0.5, then spiral movement is used, otherwise, rotational movement is used to update the whales.

3.6. Feature Selection With HHO Algorithm

The flowchart of the proposed method for feature selection using the Harris Hawks Optimization Algorithm is shown in Figure 3.3 and has the following steps:

- Size of training and test data for training and evaluation of the proposed method
- A feature vector is considered and coded as a member of Harris Hawks Optimization Algorithm
- A number of feature vectors are considered as problem solutions or members of the HHO and this population is randomly generated.
- The solutions or hawks, which are the feature vector here, are evaluated with the objective function of the problem and the most optimal feature vector is determined to detect attacks on the IoT network.
- Phases of random search, smooth encirclement, dive, etc. applied to feature vectors to be updated.
- By successive repetition of the proposed algorithm, the most optimal feature vector is updated in every iteration and the most optimal feature vector is extracted in the last iteration.

To update the feature vectors in HHO algorithm, random behavior is first performed using relation (12):

$$X(t+1) = \begin{cases} X_{rand}(t) - r_1|X_{rand}(t) - 2r_2.X(t)| & rand \geq 0.5 \\ (X_{rabbit}(t) - X_M(t)) - r_3(LB + r_4(UB - LB)) & rand < 0.5 \end{cases} \quad (3.12)$$

In this regard, $X(t)$ is the current position of a hawk or feature vectors in the current iteration or t, $X(t+1)$ is the position of a hawk or feature vectors in the new iteration, $X_{rabbit}(t)$ is the position of the most optimal solution or the most optimal feature vector and $X_{rand}(t)$ is a random position of the feature space in the problem space. Parameter $X_M(t)$ the gravity point of the population of hawks or the vectors related to the position of the cluster heads, The lower and upper bounds of the solutions in the problem space are denoted by *LB* and *UB*, respectively, while r_1, r_2, r_3 and r_4 are random values between 0 and 1. To model a smooth encirclement to update the vectors related to the position of the cluster heads, relations (13) and (14) are used in the (HHO) algorithm to update the hawks' location or feature vectors:

$$X(t+1) = \Delta X(t) - E|J.X_{rabbit}(t) - X(t)| \quad (3.13)$$

$$\Delta X(t) = X_{rabbit}(t) - X(t) \quad (3.14)$$

In these relationships, J is a random value between 0 and 2, and this parameter is used as the step of prey movement or the vectors related to the position of the optimal feature vector.



Figure 3.3. Feature selection flowchart with HHO algorithm in the proposed method

Any solution or hawk can directly reach the feature vector or dive towards it and this behavior can be defined according to relation (15):

$$X(t+1) = X_{rabbit}(t) - E|\Delta X(t)| \quad (3.15)$$

Each hawk or vectors related to it can make a decision according to the center of gathering of other hawks or feature vectors and determine their gravity point, and in this case displacement is done according to the equations of relation (16):

$$X(t+1) = X_{rabbit}(t) - E|J.X_{rabbit}(t) - X_m(t)| \quad (3.16)$$

Each hawk or its corresponding feature vector approaches the prey with the help of the average position of the population of feature vectors and the most optimal position of population or vectors related to the position of the optimal feature vector and searches around it to find the optimal feature vectors. The steps of the proposed method in feature selection using HHO algorithm are shown in Figure 3.3.

3.7. Feature Selection With JSO Algorithm

Figure 3.4 gives the flowchart of the suggested approach for feature selection utilizing the JSO Algorithm. According to the proposed flowchart for updating the feature vectors, the following three steps are performed regularly on these vectors:

- Update with movement behavior in water waves
- Update with passive motion
- Update with active and group movement

In this algorithm, the solution can move in the direction of the water waves and its direction is formulated in relation (17).

$$\overrightarrow{trend} = \frac{1}{nPop} \cdot \sum \overrightarrow{trend_i} = \frac{1}{nPop} \sum (X^* - e_c X_i) \quad (3.17)$$

In this regard, $\boldsymbol{e_c}$ is the attraction of jellyfish in the direction of water waves to find food. To move or update feature vectors in the direction of water waves, equation (18) can be used:

$$X_i\,(t+1) = X_i\,(t) + rand(0,1) \times \overrightarrow{trend} \quad (3.18)$$

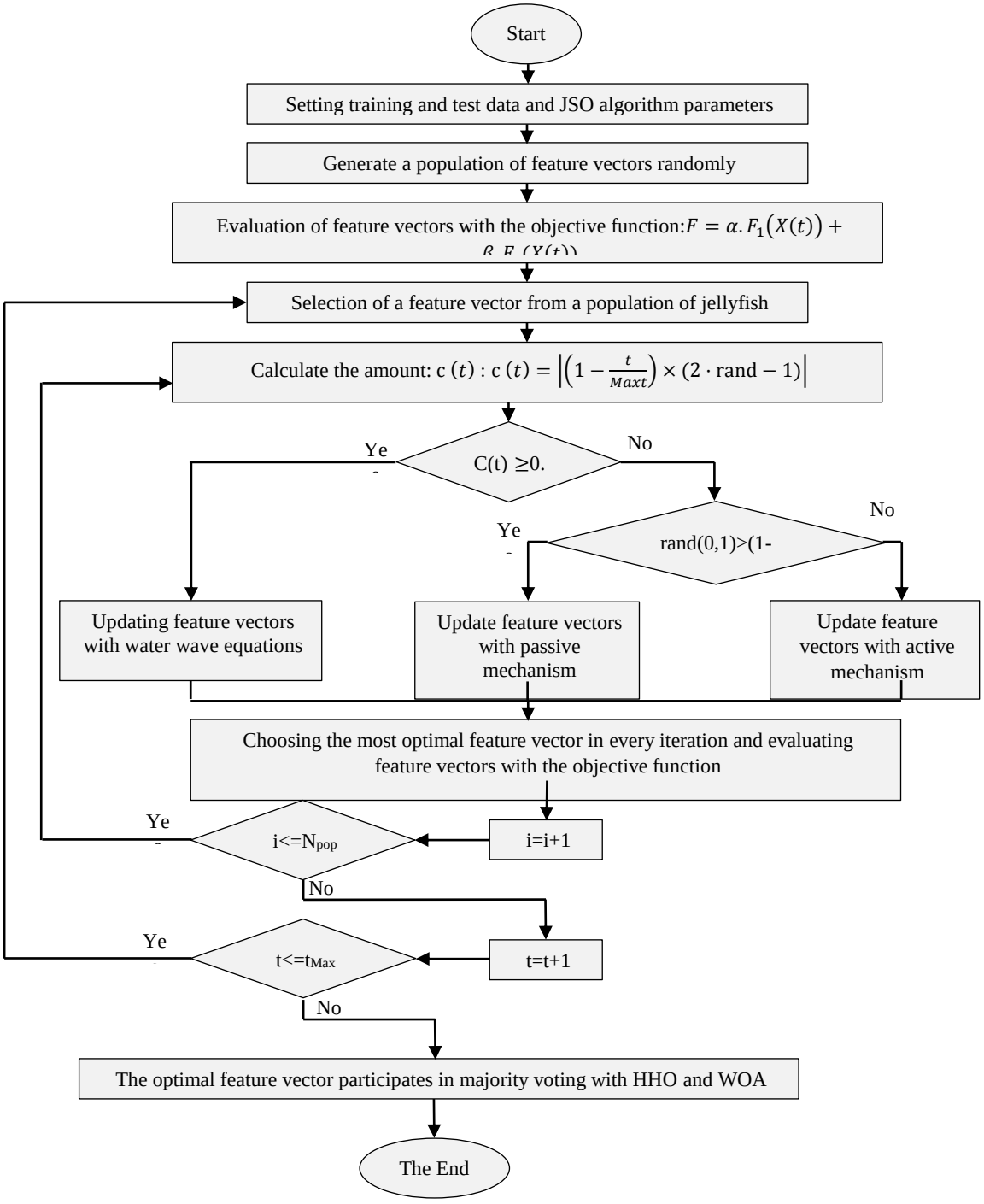


Figure 3.4. Feature selection flowchart with JSO Algorithm in the proposed method

In this relationship, a feature vector's location in the current iteration, or t, is represented by the term$X_i\ (t)$ and the new location of a feature vector is $X_i\ (t+1)$. Jellyfish Search Optimizer Algorithm can update feature vectors with passive (type A) and active (type B) behaviors. For updating with passive behavior, the equation of relation (19) can be used:

$$X_i\,(t+1) = X_i\,(t) + \gamma \cdot rand(0,1) \times (U_b - L_b) \tag{3.19}$$

In this equation, $\boldsymbol{\gamma}$ is the movement coefficient of jellyfish and its value is about 0.1, $\boldsymbol{U_b}$ is the upper limit of each dimension and L_b is the lower limit of each jellyfish. In the active mode, a feature vector such as $\boldsymbol{X_i}$ randomly selects a feature vector such as $\boldsymbol{X_j}$. In this case, if the merit of $\boldsymbol{X_i}$ is greater than $\boldsymbol{X_j}$, equation (20) is used for updating:

$$X_i\,(t+1) = X_i\,(t) + rand \cdot (X_j\,(t) - X_i\,(t)) \tag{3.20}$$

To change the search strategy, the coefficient c(t) is used and according to rule (21), it is calculated:

$$\mathrm{c}\,(t) = \left|\left(1 - \frac{t}{Maxt}\right) \times (2 \cdot \mathrm{rand} - 1)\right| \tag{3.21}$$

In this regard, $\boldsymbol{t}$ is the current counter of the algorithm and Maxt is the maximum iteration of the Jellyfish Search Optimizer Algorithm. An accidental number between 0 and 1 for each feature vector or jellyfish is generated and by comparing that with the value of C(t), the type of search is determined as shown in the flowchart in Figure 3.4.

3.8. Binarizing The Feature Vectors

In the feature selection algorithm, feature vectors need to be re-binary in each iteration, because based on the update of HHO Algorithm relationships, these feature vectors are out of binary and become continuous. To solve this challenge, the values of feature vectors can be normalized with mapping functions in the range of zero and one. Figure 3.5 shows the performance of two mapping and transformation functions V and S to normalize the feature vector. Its purpose is to have feature vectors with values from zero to one, and then based on a threshold such as 0.6 in Figure 3.6, the feature vectors are re-binary. In these diagrams, from right to left, V mapping function and S mapping function are displayed. These functions have a range between zero and one and normalize the continuous values of the feature vectors. By normalizing the feature vectors, they can be mapped to zero and one with the threshold values in Figure 3.6. In this illustration, the value of the feature vector will be

one if the house's value is greater than 0.6. Otherwise, it will be equal to zero which means feature selection.

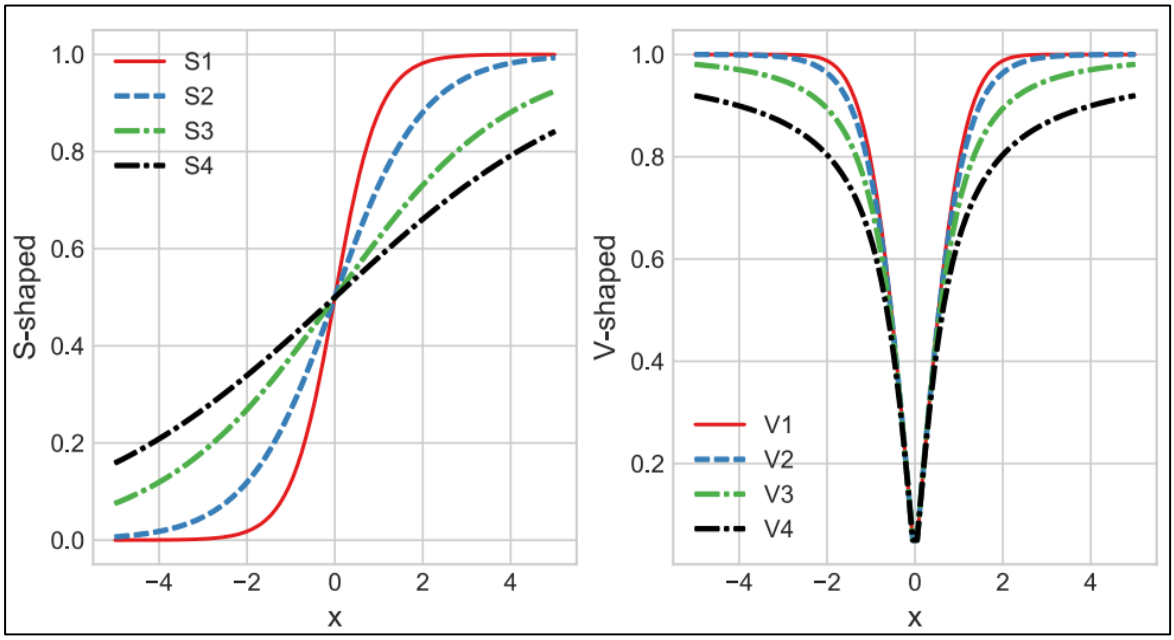


Figure 3.5. The performance of mapping functions with a range between zero and one (Mandal et al., 2020)

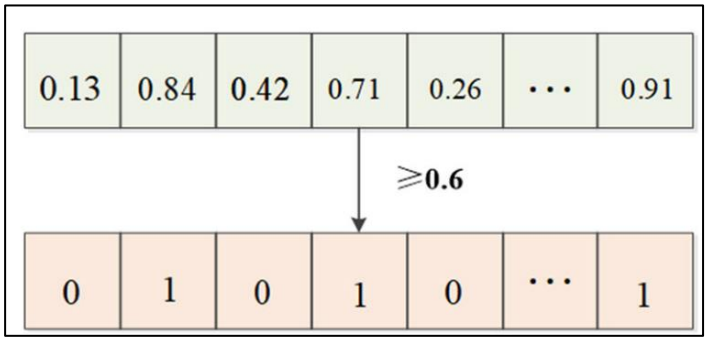


Figure 3.6. Binarizing the feature vector with thresholding (Xue & Zhao, 2022)

With changes on the feature vector, their value can become a non-zero number or one and it is necessary to make their value binary by using V and Gaussian transformation functions, which are displayed in the diagram of figure 3.5. The rules of each of the Wei and Gaussian transform functions are formulated and modeled in relation (22) and (23), respectively.

$$T(F) = \frac{1}{1+e^{-F}} \tag{3.22}$$

$$T(F) = \left|\frac{2}{\pi}\arctan(\frac{2}{\pi}F)\right| \tag{3.23}$$

In order to make each feature vector or components used in feature selection algorithms binary, the value of each feature vector component such as F_i^j can be used which is actually the feature vector i and its component according to relations (24) and (25), which respectively used the Gaussian and Wei transformation functions to make the components or feature vectors binary.

$$F_i^j = \begin{cases} 1 & rand(0,1) < T(F_i^j) \\ 0 & rand(0,1) \geq T(F_i^j) \end{cases} \tag{3.24}$$

$$F_i^j = \begin{cases} \neg F_i^j & rand(0,1) < T(F_i^j) \\ F_i^j & rand(0,1) \geq T(F_i^j) \end{cases} \tag{3.25}$$

3.9. Voting and Classification

In the proposed intrusion detection system, three algorithms WOA, HHO and JSO are used in the feature selection phase. Each of these feature selection methods output an optimal feature vector. The three optimal feature vectors that are the output of WOA, HHO, and JSO each have 41 components and the value of each component can be 0 and 1 and it respectively demonstrates the non-selection of the feature and the selection of the feature. Each feature is selected when at least two of the three feature selection algorithms WOA, HHO and JSO select it. Finally, the multi-layer artificial neural network classifier receives the optimal feature vector as shown in Figure 3.7.

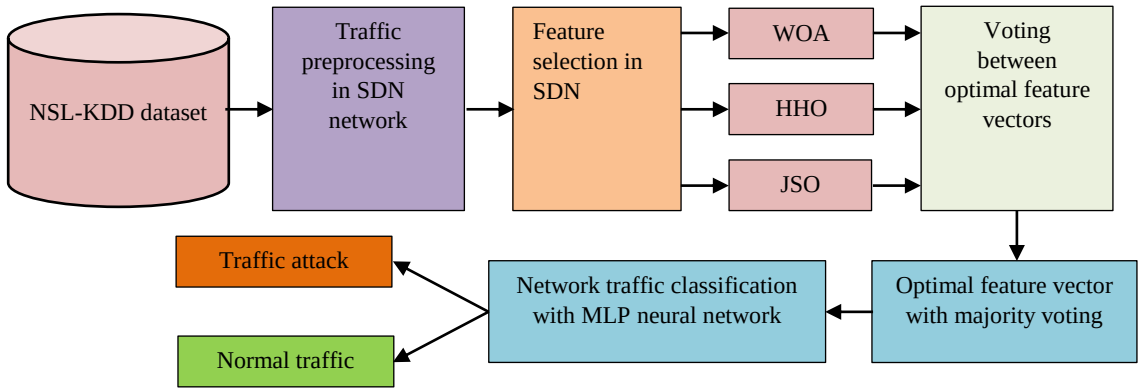


Figure 3.7. Voting and classification mechanism in the proposed intrusion detection system

In the hidden layers of an artificial neural network, there are many artificial neurons. The inputs to an artificial neuron are multiplied by the network's weights before being added with the neuron's bayes values. Bayes is represented by b, and weights are represented by w. The output of weights and Bayes are adjustable by activity functions like Sigmoid function, sigmoid tangent activity (Tanh) function, rectified activity (ReLU) function and rectified activity (Leaky ReLU) function with deviation so that the light's output does not become excessively small or large. In relation (26) and (27), the equation and rule of two sigmoid activity functions and sigmoid tangent, which are widely used in machine learning, are shown:

$$f(x) = \frac{1}{1 + exp^{-x}} \tag{3.26}$$

$$f(x) = \frac{exp^{x} - exp^{-x}}{exp^{x} + exp^{-x}} \tag{3.27}$$

Both the sigmoid activity function's range and the sigmoid tangent's range fall within the limits [0,1] and [-1,+1]. As shown by equation (28), the activity function can have an impact on the output of an artificial neural network:

$$\widetilde{f(X)} = f_{act}(W.X + b) \tag{3.28}$$

In this model, the inputs represented by $\boldsymbol{X}$ vector are multiplied by weights and Bayes and create the final output represented by $\widetilde{f(\boldsymbol{X})}$.

3.10. Data Set

To evaluate the proposed method, there are different datasets, according to the background of the studies, among the different datasets, the NSL-KDD dataset has the most use and contribution in most studies. The characteristics of the NSL-KDD data set for analyzing the proposed method are depicted in Table 3.1. The NSL-KDD dataset is a global dataset that covers various types of attacks. This dataset was compiled and prepared by a group of researchers at an American Air Force base. The KDD'99 dataset was created by DARPA in 1999 using the network traffic recorded from the 1998 dataset and later this dataset was revised and a version of the NSL-KDD dataset was presented to improve it (H. Li et al., 2019).

Table 3.1. Characteristics of NSL-KDD dataset for analysis of the suggested approach

No.	Features	Types	No.	Features	Types
1	duration	cont.	22	is_guest_login	cont.
2	protocol_type	symb.	23	count	cont.
3	service	symb.	24	srv_count	cont.
4	flag	symb.	25	serror_rate	cont.
5	src_bytes	cont.	26	srv_serror_rate	cont.
6	dst_bytes	cont.	27	rerror_rate	cont.
7	land	cont.	28	srv_rerror_rate	cont.
8	wrong_fragment	cont.	29	same_srv_rate	cont.
9	urgent	cont.	30	diff_srv_rate	cont.
10	hot	cont.	31	srv_diff_host_rate	cont.
11	num_failed_logins	cont.	32	dst_host_count	cont.
12	logged_in	cont.	33	dst_host_srv_count	cont.
13	num_compromised	cont.	34	dst_host_same_srv_count	cont.
14	root_shell	cont.	35	dst_host_diff_srv_rate	cont.
15	su_attempted	cont.	36	dst_host_same_src_port_rate	cont.
16	num_root	cont.	37	dst_host_srv_diff_host_rate	cont.
17	num_file_creations	cont.	38	dst_host_serror_rate	cont.
18	num_shells	cont.	39	dst_host_srv_serror_rate	cont.
19	num_access_files	cont.	40	dst_host_rerror_rate	cont.
20	num_outbound_cmds	cont.	41	dst_host_srv_rerror_rate	cont.
21	is_host_login	cont.			

This data set has 41 input features and one output feature that defines the normal and abnormal network traffic. The characteristics of the data set are classified into four groups: basic attributes (#1 to #9), content attributes (#10 to #22), time-based traffic attributes (#23 to #31) and host-based attributes (#32 to #41) (Choudhary & Kesswani, 2020).

A data set called NSL-KDD has been proposed to address some of the KDD'99 data set's inherent issues, which are listed in (Tavallaee et al., 2009). Due to the dearth of publicly available data sets for network-based IDSs, we believe this updated version of the KDD data set can still be used as an effective benchmark data set to aid researchers in comparing various intrusion detection techniques, despite still having some of the issues raised by McHugh.

Additionally, the NSL-KDD train and test sets have a respectable number of records. Due to this benefit, all of the data can be used for the tests instead of just a tiny sample that must be chosen at random. As a result, evaluation findings from various research projects will be comparable and consistent.

4. RESULTS AND DISCUSSION

In this chapter, we show how our work is designed.

4.1. Analysis

In this section, the proposed method based on machine learning for detecting attacks and classifying IoT traffic is implemented and the results of the research are also evaluated. In the continuation of the analysis, the proposed method is evaluated with different indicators with the results of several articles.

4.2. Evaluation Indices

Classification of images is done using the accuracy, sensitivity, specificity, and F1 indices in this study. The rules for each of these indices are introduced in relations (29), (30), and (31) respectively:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \tag{1.29}$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \tag{1.30}$$

$$precision = \frac{TN}{TN + FP} \times 100\% \tag{1.31}$$

To calculate each of these indices, True Positive (TP) integers, True Negative (TN) integers, False Positives (FP) integers and False Negative (FN) integers are utilized.

4.3. Implementation Parameters

A multilayer neural network of the MLP type has been employed in this study to assess the suggested method. The number of hidden layers is 2, 3, 4, and 5. The number of artificial neurons in the hidden layers is equal to 41 and the number of primary features of the NSL-

KDD dataset is considered. This article's implementation software is MATLAB 2021. In proposed method, the E0 and J0 coefficients in the HHO algorithm are set equal to 2, and the C coefficient in the WOA is an accidental number and is between 0 and 2. The beta coefficient is set to 3 in JSO algorithm and the omega coefficient is set to 0.1. The initial population size is set to 10 and the number of repetitions of meta-heuristic algorithms is set to 50. The normalization range is between 0 and 1 and the number of tests is set to 100.

4.4. Tests and Evaluations

To implement the suggested approach, the layers' quantities are considered equivalent to 2, 3, 4 and 5, individually and the index of accuracy, sensitivity, and specificity is calculated and analyzed for these four modes. Tables 4.1, 4.2, 4.3, and 4.4 show the effectiveness of neural network in SDN switches in four feature selection methods to detect attacks in fog nodes with increasing number of hidden layers.

Table 4.1. Changes in accuracy, sensitivity and specificity index in suggested approach by increasing the number of hidden layers with the WOA feature selection method

specificity	sensitivity	accuracy	Number of hidden layers
98.52	98.73	98.84	2
98.56	98.81	98.94	3
98.69	99.03	99.09	4
98.51	98.92	99.06	5

Table 4.2. Changes in accuracy, sensitivity, and specificity index in suggested approach by increasing the number of hidden layers with the HHO feature selection method

specificity	sensitivity	accuracy	Number of hidden layers
98.86	98.88	98.95	2
98.85	99.03	99.11	3
99.04	99.06	99.14	4
98.66	98.81	99.02	5

Table 4.3. Changes in accuracy, sensitivity and specificity index in suggested approach by increasing the number of hidden layers with the JSO feature selection method

specificity	sensitivity	accuracy	Number of hidden layers
98.77	98.81	98.84	2
98.76	98.82	99.16	3
98.92	99.12	99.17	4
98.43	99.09	99.13	5

Table 4.4. Changes in accuracy, sensitivity, and specificity index in suggested approach by increasing the number of hidden layers with feature selection method with majority voting mechanism

specificity	sensitivity	accuracy	Number of hidden layers
98.85	98.89	99.12	2
99.06	99.18	99.43	3
99.41	99.54	99.67	4
98.92	99.15	99.41	5

In Table 4.1, 4.2, 4.3 and 4.4, WOA, HHO, JSO and majority voting are selected respectively. At the same time, the number of layers of the multi-layer neural network has been changed from 2 to 5 in order to evaluate the accuracy, sensitivity, and specificity of the proposed method in detecting attacks. The evaluations show that if the number of hidden layers is chosen equal to 4, then the feature selection methods have more accuracy, sensitivity, and specificity for detecting attacks. In the graphs of figure 4.1, 4.2, and 4.3, the comparison between four feature selection methods for feature selection in accuracy, sensitivity and specificity indices with hidden layer is displayed.

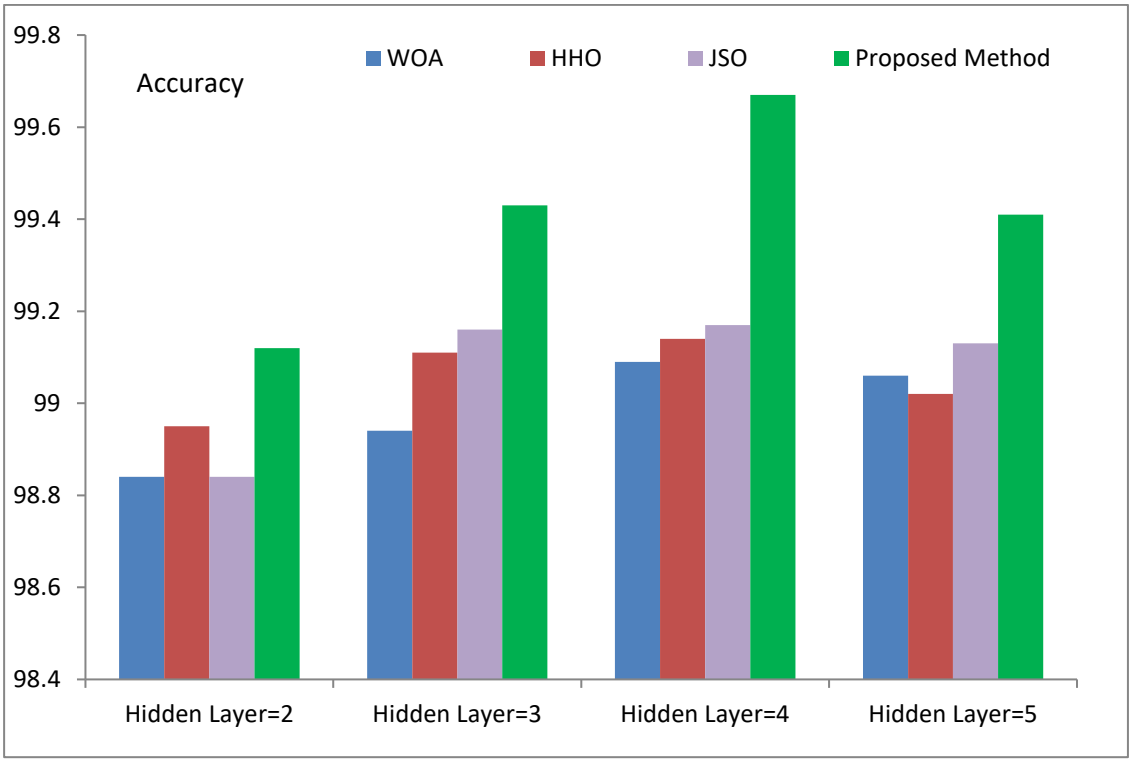


Figure 4.1. Comparison of accuracy index of feature selection methods

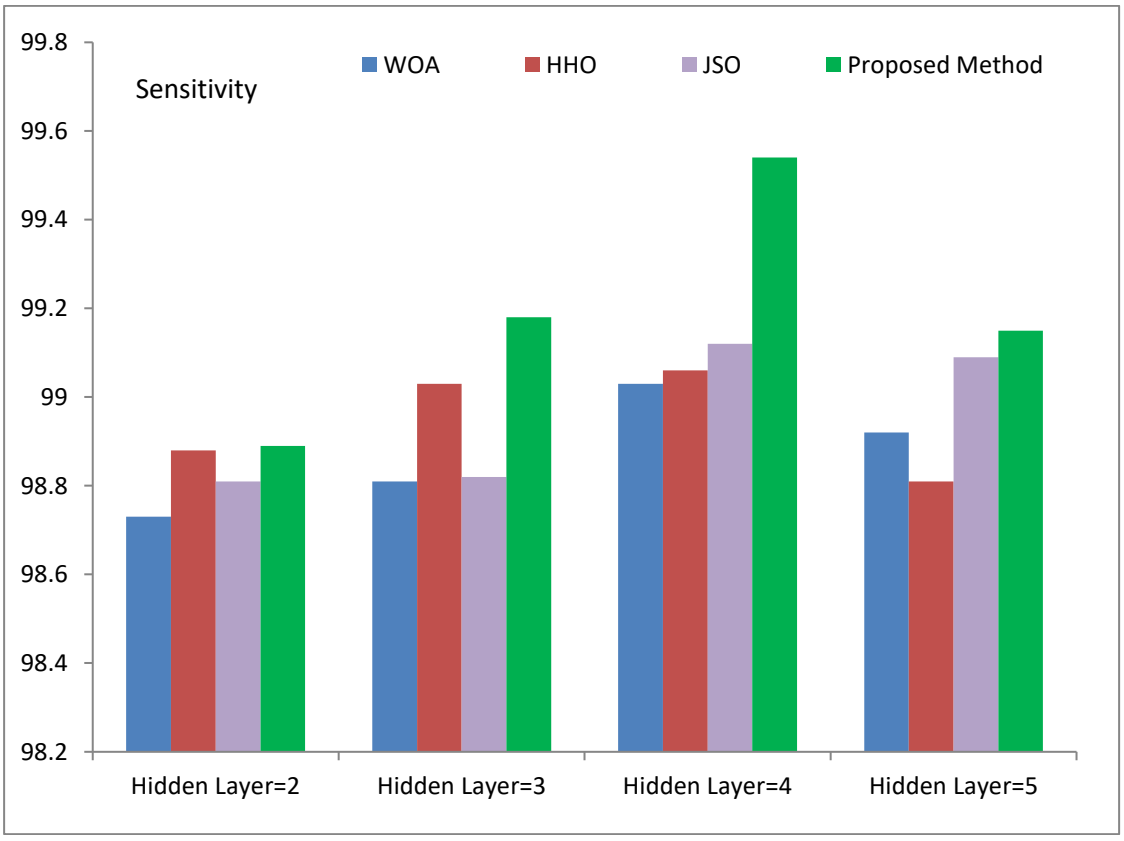


Figure 4.2. Comparison of the sensitivity index of feature selection methods

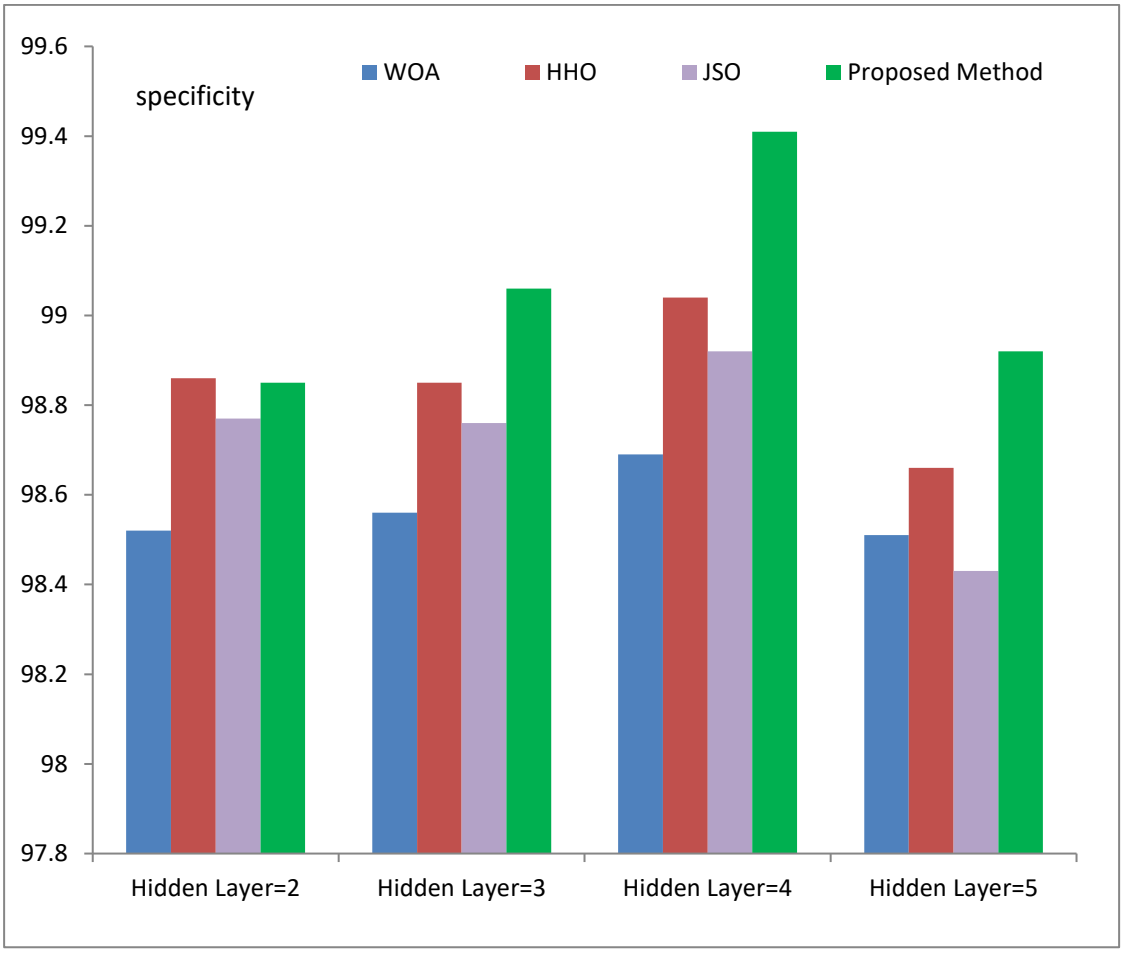


Figure 4.3. Comparison of accuracy index of feature selection methods

The analysis of accuracy, sensitivity and specificity charts shows that the proposed algorithm, which has a majority voting mechanism, is more accurate and efficient than the WOA, HHO and JSO feature selection methods in detecting network intrusion. By increasing the number of hidden layers to 4, the efficiency of the neural network will increase. In other words, if the multilayer neural network has 4 layers, then its accuracy in detecting attacks in the SDN network is the highest. The evaluations show that if the number of hidden layers in the proposed approach is considered equal to 2, the accuracy, sensitivity and specificity of suggested approach are 99.12%, 98.89% and 98.85% respectively. If the quantity of the hidden layers is equivalent to 3, the accuracy, sensitivity and specificity of the suggested approach are 99.43%, 99.18% and 99.06% respectively. If the quantity of the hidden layers is considered as 4, the accuracy, sensitivity and specificity of the suggested approach are 99.67%, 99.54% and 99.41% respectively. By increasing the number of layers to 5, the accuracy, sensitivity, and specificity of the suggested approach are 99.41%, 99.15%, and 98.92%, respectively, and it does not show an increase in accuracy. In the

following, the proposed method by filter feature selection method with support vector machine classifier[1], warper with support vector machine classifier[2] and combined with support vector machine classifier[3], filter feature selection method with nearest neighbor[4], warper with nearest neighbor[5] and a combination with the nearest neighbor classifier[6], a filter feature selection method with a neural network[7], Warper with a neural network[8] and a combination with a neural network classifier[9] and a filter feature selection method with a Bayesian network[10], a Warper with a Bayesian network[11] and a combination with classification Bayesian network generator[12] has been compared.

Here, the proposed method is compared with the results of the research (Kim et al., 2022), which was presented in 2022. In the diagram of Figure 4.4, the proposed method is compared with multiple classifiers in combination with feature selection.

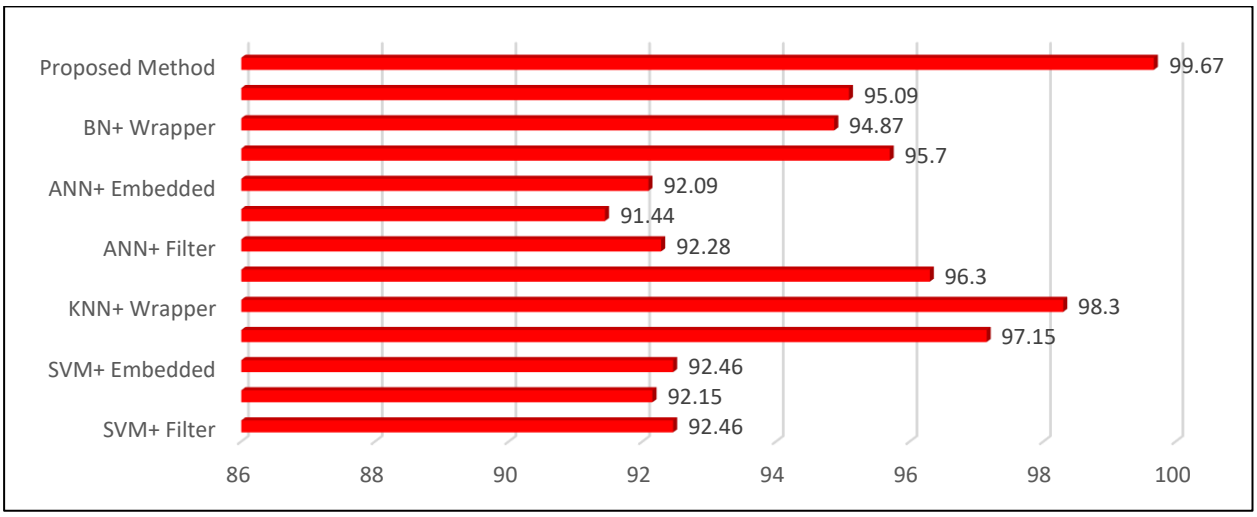


Figure 4.4. Comparison of attack detection accuracy in the proposed method with feature selection methods

[1] SVM+ Filter
[2] SVM+ Wrapper
[3] SVM+ Embedded
[4] KNN+ Filter
[5] KNN+ Wrapper
[6] KNN+ Embedded
[7] ANN+ Filter
[8] ANN+ Wrapper
[9] ANN+ Embedded
[10] BN+ Filter
[11] BN+ Wrapper
[12] BN+ Embedded

The evaluations show that the accuracy of the proposed method, which is equal to 99.67%, is more accurate than other methods of machine learning and feature selection. The accuracy of detecting attacks in the filter feature selection method with support vector machine classifier, warper with support vector machine classifier and combination with support vector machine classifier is 92.46%, 92.15% and 92.46%, respectively. The accuracy of the feature selection method of nearest neighbor filter, nearest neighbor warper and a combination with nearest neighbor classifier is 97.15%, 98.30% and 96.30% respectively. The feature selection method of filter with neural network, warper with neural network and combination with neural network classifier in detecting attacks has an accuracy of 92.28%, 91.44% and 92.09% respectively. The accuracy of the filter feature selection method with Bayesian network, warper with Bayesian network and combination with Bayesian network classifier in detecting attacks is 95.70%, 94.87% and 95.09%, respectively. The proposed method has also been compared and analyzed with three advanced feature selection methods in the article (Mishra et al., 2022), which was presented in 2022, in two indices of accuracy and sensitivity and these comparisons are displayed in the graph of figure 4.5 and 4.6. The evaluations show that the accuracy of detecting attacks in PBC, GA and PSO methods is 99.15%, 97.44% and 91.19%, respectively, while the accuracy of the proposed method is 99.67% and it is more accurate than these methods. The evaluations show that the sensitivity of PBC, GA, and PSO methods is 99.10%, 96.55%, and 95.55% respectively and the sensitivity for the suggested approach in discovering attacks is 99.54%, which is higher than these methods.

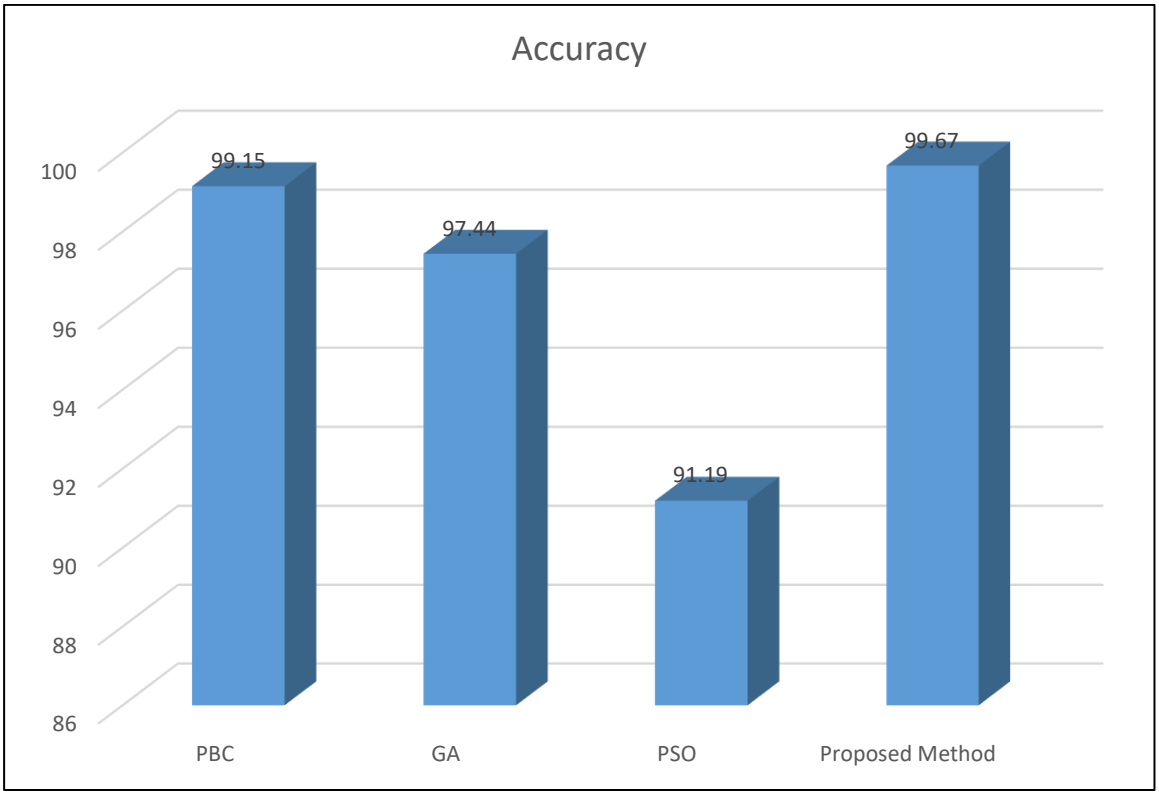


Figure 4.5. Comparison of the accuracy index of the proposed method with feature selection methods with optimization approach

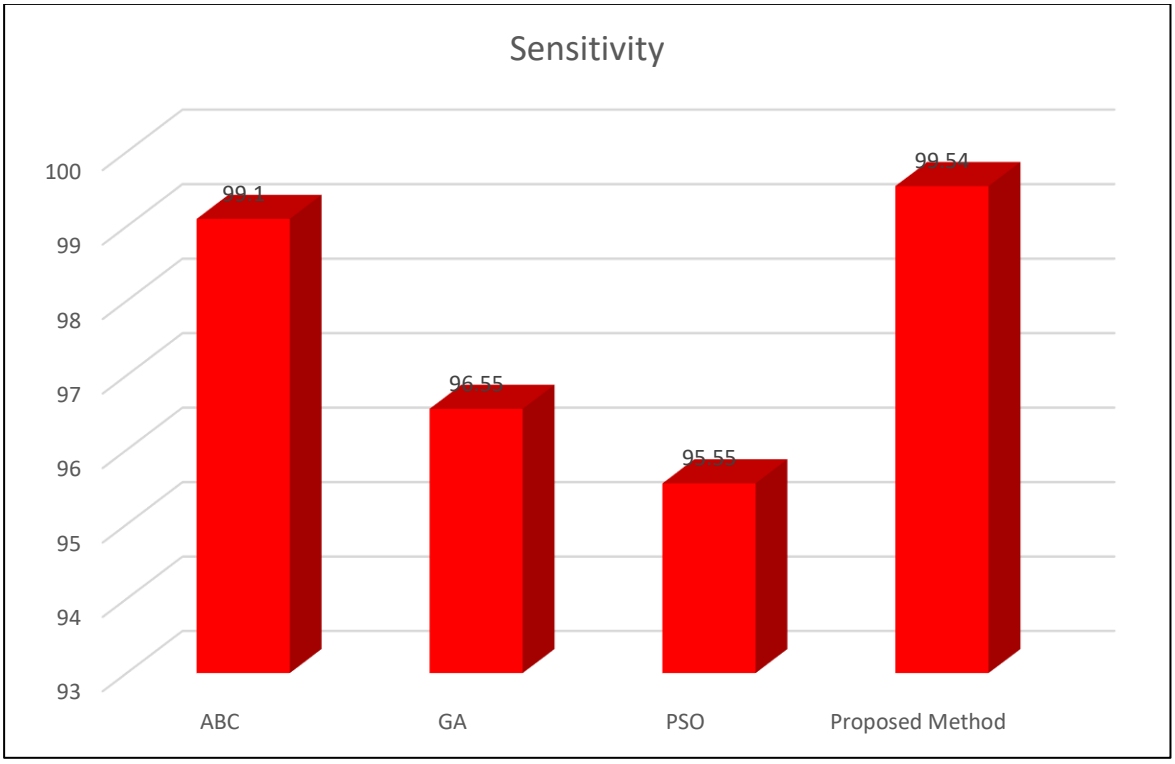


Figure 4.6. Comparison of the sensitivity index of the proposed method with the feature selection methods with the optimization approach

In this thesis, all scenario has been run about 20 times, and the standard deviation, mean absolute error, and mean average values has been calculated. This index has been shown in figure 4.7.

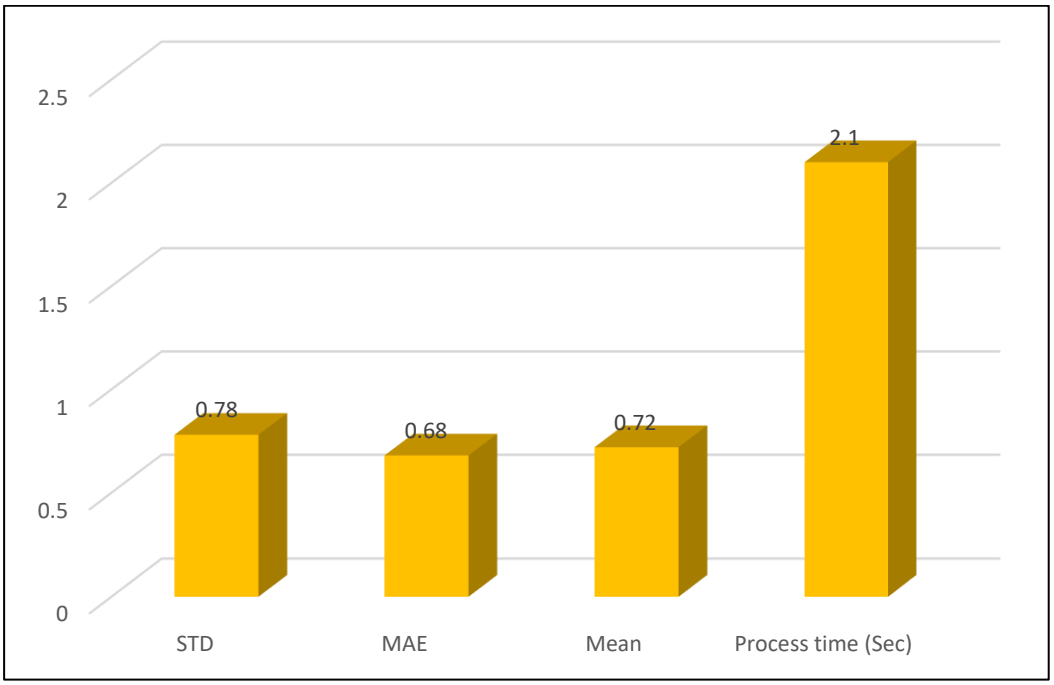


Figure 4.7. STD, MAE, Mean, and process time

5. CONCLUSION AND FUTURE WORKS

5.1. Conclusion

The IoT network has many nodes and for this reason, its data and traffic are so large that it is of the big data type. Therefore, analyzing this traffic requires a lot of time. Cyber-attacks are one of the important challenges of all types of networks, including the IoT network, which can disrupt the infrastructure of this network. For this reason, it is necessary to use security systems such as antivirus and firewall to deal with this type of attacks. Network intrusion detection system is one of the important methods to improve network security and it can analyze network traffic and detect any network intrusion and usually to create these systems from Knowledge discovery methods such as machine learning and data mining methods can be used to analyze network traffic and detect network intrusion. The proposed method to detect network attacks uses SDN architecture. In the proposed method, unlike other researches, in the feature selection phase, majority voting is used so that the selected features are selected with great accuracy. Each SDN switch in the proposed method plays the role of an intrusion detection system specified by optimal feature vector controller with the help of the majority voting mechanism. In the feature selection phase, three methods of Harris Hawks Optimization, whale optimization, and jellyfish optimization are used for feature selection. An important feature is recognized when it is selected by two or three feature selection methods with a majority voting mechanism. The optimal feature vector can be utilized to train a multilayer neural network in SDN switches in second phase. Experiments and evaluation on NSL-KDD dataset show that using 4 hidden layers will get better results. The evaluations showed that the accuracy, sensitivity, and specificity of the proposed method are 99.67%, 99.54%, and 99.41%, respectively, and are more accurate than WOA, HHO, and JSO feature selection algorithms. According to the experiments the accuracy of the suggested approach in detecting attacks is higher comparing to PBC, GA and PSO methods. The proposed approach has the benefit of being more accurate at detecting attacks comparing to similar meta-heuristic algorithms. Another advantage of the proposed method is to reduce the feature selection error with the majority voting mechanism. The challenge of the proposed method in attack detection has low process time for feature selection compared to WOA, HHO and JSO feature selection algorithm. The use of deep learning and other data sets in intrusion detection is one of our future proposals.

5.2. Future Work

In future work, we can use the other metaheuristic method such as the particle swarm optimization, ant colony optimization.

REFERENCES

Alkasassbeh, M., Al-Naymat, G., Hassanat, A., Almseidin, M. (2016). Detecting distributed denial of service attacks using data mining techniques. *International Journal of Advanced Computer Science and Applications*, 7(1), 436–445.

Alzahrani, M. Y., Bamhdi, A. M. (2022). Hybrid deep-learning model to detect botnet attacks over internet of things environments. *Soft Computing*, 1–15.

Arul, E. (2021). Deep nonlinear regression least squares polynomial fit to detect malicious attack on IoT devices. *Journal of Ambient Intelligence and Humanized Computing*, 12(1), 769–779.

Aktemur, I., Erensoy, K., Kocyigit, E. (2020). Optimization of Waste Collection in Smart Cities with the use of Evolutionary Algorithms. *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 1–8.

Bhushan, B., Sahoo, G. (2020). Requirements, protocols, and security challenges in wireless sensor networks: An industrial perspective. In *Handbook of computer networks and cyber security* (pp. 683–713). Springer.

Chalé, M., Bastian, N. D. (2022). Generating realistic cyber data for training and evaluating machine learning classifiers for network intrusion detection systems. *Expert Systems with Applications*, 207, 117936.

Chou, J.-S., Truong, D.-N. (2020). Multiobjective optimization inspired by behavior of jellyfish for solving structural design problems. *Chaos, Solitons & Fractals*, 135, 109738.

Chou, J.-S., Truong, D.-N. (2021). A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean. *Applied Mathematics and Computation*, *389*, 125535.

Choudhary, S., Kesswani, N. (2020). Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT. *Procedia Computer Science*, 167, 1561–1573.

Chakravarty, S. (2020). Feature selection and evaluation of permission-based Android malware detection. *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)*(48184), 795–799.

Chopra, K., Gupta, K., Lambora, A. (2019). Future internet: The internet of things-a literature review. *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, 135–139.

Carrez, F. (2013). *Converged architectural reference model for the IoT*.

Dhieb, N., Ghazzai, H., Besbes, H., Massoud, Y. (2020). Scalable and Secure Architecture for Distributed IoT Systems. *2020 IEEE Technology & Engineering Management Conference (TEMSCON)*, 1–6.

Domínguez-Limaico, H., Nicolalde Quilca, W., Zambrano, M., Cuzme-Rodríguez, F., Maya-Olalla, E. (2023). Intruder Detection System Based Artificial Neural Network for Software Defined Network. *International Conference of Technological Research*, 315–328.

Dora, V., Lakshmi, V. N. (2022). Optimal feature selection with CNN-feature learning for DDoS attack detection using meta-heuristic-based LSTM. *International Journal of Intelligent Robotics and Applications*, 6(2), 323–349.

Emine, B. A. Ş., & Ülker, E. (2020). An efficient binary social spider algorithm for feature selection problem. *Expert Systems with Applications*, 146, 113185.

Eskandari, M., Janjua, Z. H., Vecchio, M., Antonelli, F. (2020). Passban IDS: An intelligent anomaly-based intrusion detection system for IoT edge devices. *IEEE Internet of Things Journal*, 7(8), 6882–6897.

Friedemann, M., Floerkemeier, C. (2010). From the Internet of Computers to the Internet of Things. *Informatik-Spektrum*, 33(2), 107–121.

Gu, H., Lai, Y., Wang, Y., Liu, J., Sun, M., Mao, B. (2022). DEIDS: a novel intrusion detection system for industrial control systems. *Neural Computing and Applications*, 1–19.

Guezzaz, A., Benkirane, S., Azrour, M. (2022). A Novel Anomaly Network Intrusion Detection System for Internet of Things Security. In *IoT and Smart Devices for Sustainable Environment* (pp. 129–138). Springer.

Gupta, B. B., Quamara, M. (2020). An overview of Internet of Things (IoT): Architectural aspects, challenges, and protocols. *Concurrency and Computation: Practice and Experience*, 32(21), e4946.

Haddadpajouh, H., Mohtadi, A., Dehghantanaha, A., Karimipour, H., Lin, X., Choo, K.-K. R. (2020). A multikernel and metaheuristic feature selection approach for IoT malware threat hunting in the edge layer. *IEEE Internet of Things Journal*, 8(6), 4540–4547.

Hadi, M. R., Mohammed, A. S. (2022). A novel approach to network intrusion detection system using deep learning for Sdn: Futuristic approach. *ArXiv Preprint ArXiv:*2208.02094.

Haque, M. R., Tan, S. C., Yusoff, Z., Nisar, K., Kaspin, R., Haider, I., Nisar, S., Rodrigues, J. J. P. C., Shankar Chowdhry, B., & Uqaili, M. A. (2022). Unprecedented smart algorithm for uninterrupted SDN services during DDoS attack. *Computers, Materials & Continua*, 70(1), 875–894.

Harada, R., Shibata, N., Kaneko, S., Honda, K., Terada, J., Ishida, Y., Akashi, K., & Miyachi, T. (2022). Quick Suppression of DDoS Attacks by Frame Priority Control in IoT Backhaul with Construction of Mirai-based Attacks. *IEEE Access*, 10, 22392–22399.

Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, 97, 849–872.

Hosseini, S., Sardo, S. R. (2022). Network intrusion detection based on deep learning method in internet of thing. *Journal of Reliable Intelligent Environments*, 1–13.

Huo, R., Zeng, S., Wang, Z., Shang, J., Chen, W., Huang, T., Wang, S., Yu, F. R., Liu, Y. (2022). A comprehensive survey on blockchain in industrial internet of things: Motivations, research progresses, and future challenges. *IEEE Communications Surveys & Tutorials*.

Imran, M., Khan, S., Hlavacs, H., Khan, F. A., Anwar, S. (2022). Intrusion detection in networks using cuckoo search optimization. *Soft Computing*, 1–13.

Jarial, S. (2022). Internet of Things application in Indian agriculture, challenges and effect on the extension advisory services–a review. *Journal of Agribusiness in Developing and Emerging Economies*.

Khammassi, C., Krichen, S. (2017). A GA-LR wrapper approach for feature selection in network intrusion detection. *Computers & Security*, 70, 255–277.

Kharkwal, A., Mishra, S., Paul, A. (2022). Cross-Layer DoS Attack Detection Technique for Internet of Things. *2022 7th International Conference on Communication and Electronics Systems (ICCES)*, 368–372.

Khowaja, S. A., Khuwaja, P. (2021). Q-learning and LSTM based deep active learning strategy for malware defense in industrial IoT applications. *Multimedia Tools and Applications*, 80(10), 14637–14663.

Kim, Y.-E., Kim, Y.-S., Kim, H. (2022). Effective Feature Selection Methods to Detect IoT DDoS Attack in 5G Core Network. *Sensors*, 22(10), 3819.

Kranthi, S., Kanchana, M., Suneetha, M. (2022). A Study of IDS-based Software-defined Networking by Using Machine Learning Concept. In *Advances in Data and Information Sciences* (pp. 65–79). Springer.

Liu, H., Sun, Y., Kim, M. S. (2011). Fine-grained DDoS detection scheme based on bidirectional count sketch. *2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*, 1–6.

Le, K.-H., Nguyen, M.-H., Tran, T.-D., Tran, N.-D. (2022). IMIDS: An intelligent intrusion detection system against cyber threats in IoT. *Electronics*, 11(4), 524.

Le, T.-T.-H., Kim, H., Kang, H., Kim, H. (2022). Classification and Explanation for Intrusion Detection System Based on Ensemble Trees and SHAP Method. *Sensors*, 22(3), 1154.

Li, H., Wei, F., Hu, H. (2019). Enabling dynamic network access control with anomaly-based IDS and SDN. *Proceedings of the ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*, 13–16.

Li, R., Li, Q., Huang, Y., Zhang, W., Zhu, P., Jiang, Y. (2022). IoTEnsemble: Detection of Botnet Attacks on Internet of Things. *European Symposium on Research in Computer Security*, 569–588.

Liu, K., Xu, S., Xu, G., Zhang, M., Sun, D., Liu, H. (2020). A review of android malware detection approaches based on machine learning. *IEEE Access*, 8, 124579–124607.

Liu, Z., Thapa, N., Shaver, A., Roy, K., Yuan, X., Khorsandroo, S. (2020). Anomaly detection on iot network intrusion using machine learning. *2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (IcABCD)*, 1–5.

Luo, X., Li, J., Wang, W., Gao, Y., Zhao, W. (2021). Towards improving detection performance for malware with a correntropy-based deep learning method. *Digital Communications and Networks*, 7(4), 570–579.

Machine, C. (2014). *The Only Coke Machine on the Internet*. The Carnegie Mellon University Computer Science Department, Jan.

Mandal, A. K., Sen, R., Chakraborty, B. (2020). Analysis of various transfer functions for binary owl search algorithm in feature selection problem. *International Journal of Applied Science and Engineering*, 17(3), 281–297.

Mirjalili, S., Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, 95, 51–67.

Mishra, N., Singh, R. K., Yadav, S. K. (2022). Detection of DDoS Vulnerability in Cloud Computing Using the Perplexed Bayes Classifier. *Computational Intelligence and Neuroscience*, 2022.

Munirathinam, S. (2020). Industry 4.0: Industrial internet of things (IIOT). In *Advances in computers* (Vol. 117, Issue 1, pp. 129–164). Elsevier.

Omolara, A. E., Alabdulatif, A., Abiodun, O. I., Alawida, M., Alabdulatif, A., Arshad, H. (2022). The internet of things security: A survey encompassing unexplored areas and new insights. *Computers & Security*, 112, 102494.

Putra, G. D., Dedeoglu, V., Kanhere, S. S., Jurdak, R. (2020). Towards scalable and trustworthy decentralized collaborative intrusion detection system for iot. *2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI)*, 256–257.

Palermo, F. (2014). Internet of Things done wrong stifles innovation. *Information Week*, 7.

Prabakaran, S., Ramar, R., Hussain, I., Kavin, B. P., Alshamrani, S. S., AlGhamdi, A. S., Alshehri, A. (2022). Predicting attack pattern via machine learning by exploiting stateful firewall as virtual network function in an SDN network. *Sensors*, 22(3), 709.

Pundir, S., Obaidat, M. S., Wazid, M., Das, A. K., Singh, D. P., Rodrigues, J. J. P. C. (2021). MADP-IIME: malware attack detection protocol in IoT-enabled industrial multimedia environment using machine learning approach. *Multimedia Systems*, 1–13.

Raju, A. D., Abualhaol, I. Y., Giagone, R. S., Zhou, Y., Huang, S. (2021). A survey on cross-architectural iot malware threat hunting. *IEEE Access*, *9*, 91686–91709.

Stolfo, S. J., Fan, W., Lee, W., Prodromidis, A., Chan, P. K. (2000). Cost-based modeling for fraud and intrusion detection: Results from the JAM project. *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00*, 2, 130–144.

Samie, F., Bauer, L., Henkel, J. (2019). From cloud down to things: An overview of machine learning in internet of things. *IEEE Internet of Things Journal*, 6(3), 4921–4934.

Schull, W. J., Otake, M., Neel, J. V. (1981). Genetic effects of the atomic bombs: a reappraisal. *Science*, 213(4513), 1220–1227.

Sokkalingam, S., Ramakrishnan, R. (2022). An intelligent intrusion detection system for distributed denial of service attacks: A support vector machine with hybrid optimization algorithm based approach. *Concurrency and Computation: Practice and Experience*, 34(27), e7334.

Song, S., Wang, P., Heidari, A. A., Wang, M., Zhao, X., Chen, H., He, W., Xu, S. (2021). Dimension decided Harris hawks optimization with Gaussian mutation: Balance analysis and diversity patterns. *Knowledge-Based Systems*, 215, 106425.

Su, S., He, C., Xu, L. (2022). Quasi-Reflective Chaotic Mutant Whale Swarm Optimization Fused with Operators of Fish Aggregating Device. *Symmetry*, 14(4), 829.

Thirimanne, S. P., Jayawardana, L., Yasakethu, L., Liyanaarachchi, P., Hewage, C. (2022). Deep Neural Network Based Real-Time Intrusion Detection System. *SN Computer Science*, 3(2), 1–12.

Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 1–6.

Wu, Y.-J., Hwang, P.-C., Hwang, W.-S., Cheng, M.-H. (2020). Artificial intelligence enabled routing in software defined networking. *Applied Sciences*, 10(18), 6564.

Xue, Y., Zhao, Y. (2022). Structure and weights search for classification with feature selection based on brain storm optimization algorithm. *Applied Intelligence*, 52(5), 5857–5866.



Gazili olmak ayrıcalıktır