DEVELOPMENT OF ARTIFICIAL NEURAL NETWORK BASED BACKDOOR ATTACK DETECTION TECHNIQUE WITH BINARY PARTICLE SWARM OPTIMIZATION FOR FEATURE SELECTION

BY

LAWAL, ABDULLAHI M.
MTech/SICT/2017/7234

DEPARTMENT OF CYBER SECURITY TECHNOLOGY

FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA

SEPTEMBER, 2021

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

1.0 INTRODUCTION

1.1 Background to the Study

Nowadays, computers play a significant role in all facet of human life. The spectrum of security issues considers of content of our data and information. Hence, working in a secure cyber world has a dramatic impact on users or organize s privacy. Security threats come from different sources such as natural forces and people known as unauthorized users (Salimi & Arastouie, 2016). Most of the time, an unauthorized access is run by using a special malicious software called “malware”.

Backdoor can be defined as an intentional construct inserted into a system known to the system implementer unknown to the end user, that serve to compromise its perceived security (Thomas *et al*., 2018). In the last ten years, malware attacks have become a common crime story online. Nowadays, well-known threats, including viruses, worms, trojans, backdoors, exploits, password stealers, and spyware, have reached millions, and among these threats, the backdoor attack has a high rate of intrusion across global networks around the world (Microsoft, 2012). Binary particle swarm optimization (BPSO) is one of the metaheuristic optimization methods, the algorithm has been constructed entirely on randomness (Arican & Polat, 2020). Malware can be described as various types of software, which have the capacity to wreak havoc on a computer system or illegally make use of this information without the consent of the users (Dada *et al.*, 2019).

Backdoors are a method of bypassing authentication or other security controls in order to access a computer system or the data contained on that system. Backdoors can exist at the system level, in a cryptographic algorithm, or within an application (Chris *et al*., 2016).

Any mechanism that bypass a normal security check; it may allow unauthorized access to functionality in a program or onto a compromised system. A backdoor may take the form of a hidden part of a program, a separate program or may be a hardware feature (Chang *et al.*, 2018).

The backdoor attack is a hidden technique used to gain remote access to a machine or another system without authentication. It was a major threat in recent years and is one of the threats that cause serious concerns because the outbound it generates consists of several types of packages and exerts dangerous control over a range of hosts (Khalid and Mohd, 2016). As such, detecting backdoors has become an urgent demand today. Feature selection is an effective way of reducing the number of features of data, which improve the performance of classification in machine learning (Ji *et al.*, 2020).

Artificial neural network refers to a computational model simulating the way a neural networks in the brain gives practical techniques to learning continuous, categorical and vector-valued functions from examples (Rad, 2018). ANN is a parallel-distributed system, which acquire a very different system from the olden artificial intelligence and information-processing techniques defeat the shortcomings of the olden logic-based artificial intelligence in taking care of intuition and unstructured information. It has the edge of adaptable, self-organizing and real-time learning (Wu &Feng, 2017).

Machine learning that has been implemented in backdoor detection, supervised method that entails classification or prediction of problems in other to indicate the hidden association between the target class and independent variable are known to be employed in data mining. For supervised learning, classifiers permit tagging of an attribute to observation, this is to ensure the classification of data not observed on the training data.

A backdoor detection system is developed with the implementation of classification algorithms to distinguish between legitimate and backdoor attack.

1.2 Statement of the Research Problem

In the last decade, backdoor detection is a challenging issue, especially when it comes to automatic detection. Most of the detection techniques currently used for backdoor involves manual process, which are laborious in native (Thomas *et al*., 2018). Undoubtedly, all computer viruses are undesirable, but backdoor viruses are especially dangerous because they can bypass normal authentication systems and use a hidden technique that allows a remote attacker to access and forward a user's personal information (Choi and Cho, 2012). Backdoor attack in Convolutional Neural Network are the focus study points of view and improvements has been seen in term of attack defense while backdoor get little consideration in RNN(Dai *et al.,* 2019). Several backdoor attack detection models have been proposed. Due to the high dimension of backdoor dataset, most of these models focuses more on reducing the dimension of the backdoor datasets because the quality and number of features (Salimi & Arastouie, 2011) . Affect performance of machine learning models. Filter-based feature selection techniques were mostly deployed due to their fastness and ease of use. However, they are often less accurate than the wrapper-based feature selection methods. Therefore, this research proposes the detection of backdoor attacks using binary particle swarm optimization and artificial neural network for improved feature selection and backdoor attacks detection respectively.

1.3 Aim and Objectives of the Study

The Aim of this research is to develop Artificial Neural Network Based backdoor attack detection technique with Binary Particle Swarm Optimization for feature selection.

The objectives of the study are to:

i. Formulate a binary particle swarm optimization (BPSO) feature selection algorithm for backdoor dimensionality reduction.

ii. Develop Artificial Neural Network (ANN) based backdoor attack detection model using the selected features in (i).

iii. Evaluate Performance of the model developed (ii) using Accuracy, False positive rate, Precision, Recall, F-measure, Error rate.

iv. Validate the developed model against other machine learning models which includes; Bayes Net, Bayesian LR, NaiveBayes, SVM, Kstar. Stacking, Threshold Selection, Randomizable Filter and Zero R.

1.4 Scope of the Study

This research is subjected to developing of a model using Artificial Neural Network (ANN) and Binary Particle Swarm Optimization for Backdoor Attack detection and improved feature selection, respectively. This work is limited to detecting backdoor attack using existing backdoor dataset.

1.5 Significance of the Study

This study would help to develop a robust system that can detect backdoor attack with high accuracy and low false positive rate. This research will help in reducing the high of data theft, stop website defacing and help in detecting the inserted backdoors in a computer or servers. The system manufacturers, system user's, financial institutions and other ICT sectors can make use of this research work to detect and protect against backdoor attacks or backdoors that may be inserted in a system, website or server intentionally or accidentally.

CHAPTER TWO

2.0 LITERATURE REVIEW

2.1 Backdoor

Backdoor is a class of malware that offers a supplementary stealthy “entrance” to the system for attackers. The backdoor itself does not directly harm the system but it opens the door for attackers to wreak havoc. Due to this characteristic, backdoors are in no way used individually. Ordinarily, a backdoor is antecedent malware attack or other forms of attacks (Dada *et al.,* 2019). A hidden technique is used for getting remote access to a machine or other system that without authentication Intrusion: An illegal act of entering to a computer or network or any system (Khalid and Mohd, 2016).

2.2 Types of Backdoor

2.2.1 System backdoors

System backdoors are kind of backdoors, which use a vulnerability to establish ongoing root access to a system. At the system level, system backdoors are usually created by an intruder for his/her future access to the system even if the past vulnerabilities are remediated (Salimi and Arastouie, 2016). System backdoors are backdoors that allow access to data and processes at the system level. Rootkits, remote access software, and deliberate system misconfiguration by an attacker fall into this category. An attacker who has compromised a system so that system access can be retain even if the vulnerability they used to gain initial access is remediated typically creates system backdoors. Malware such as remote access trojans or “bots” specifically created to compromise a system also fall into the system backdoor category. This malware can be installed through a vulnerability or social engineering (Chris *et al*., 2016)

2.2.2 Application backdoors

Application backdoors are illegitimate software mainly amid to be installed on a system in order to bypass its security controls. These types of backdoors can be discussed in three categories:

i. Special credential backdoor, the most common, a privileged account known only to those who designed it and those whom they share it with.
ii. Malicious backdoors

    The ones planted by programmers who intend to do harm or are paid by those who intend to do harm.
iii. Support backdoors

    The one left intentionally for support staff to gain easy access to an application for trouble shooting (Salimi and Arastouie, 2016).

Application backdoors as versions of legitimate software modified to bypass security mechanisms under certain conditions. These legitimate programs are meant to be installed and running on a system with the full knowledge and approval of the system operator. Application backdoors can result in the compromise of the data and transactions performed by an application. They can also result in system compromise (Chris and Chris, 2016).

Application backdoors are often inserted in the code by someone who has legitimate access to the code. Other times the source code or binary to an application is modified by someone who has compromised the system where the source code is maintained or the binary is distributed. Another method of inserting an application backdoor is to subvert the compiler, linker, or other components in the development tool chain used to create the application binary from the source code (Thomas *et al*., 2018 ; Chris *et al*., 2016).

2.2.3 Crypto backdoors

Crypto backdoors are deliberately designed for special keys or messages that allow an intruder's accessibility to clear text messages that they should not, in other words are portals, which are lightly encrypted and are easy to break through often used in the hacking world (Salimi and Arastouie, 2016). Crypto backdoors are intentionally designed weaknesses in a cryptosystem for particular keys or messages that allow an attacker to gain access to clear-text messages that they should not (Chris *et al*., 2016).

2.3 Backdoor Attack

Backdoor attacks are of different kind of attack, such that the adversary chooses a trigger, creates some poisoned data based on the trigger, and gives it to the target user to train a deep model with. The most common backdoor attack develops poisoned data by pasting the trigger on the original data and modifying their label to the target group (Saha *et al.,* 2019). There are two ways of defending against backdoor attack; Outlier detection and Reverse trigger. The Outlier normalize the backdoor bits by removing out-of-sample items from the average distribution for each, while Reverse trigger detects by the use of a normal data classification output with a random trigger known by the model (Kwon, 2020)

Sub7 is the first backdoor code that was created in 1990 which enabled any user backdoor (hacker) to get access to a victim's computer (Decloedt and Van Heerden, 2017). According to the mentioned study, programmers have used the backdoors as tools for many years to check and debug applications, this is generally preferred when a programmer (developer) is programming or improving a software application that needs using authentication in order to test and run the application. This software (backdoor) becomes a big threat once dishonest software discovers and using them to gain illegal

access to the victim's applications. Backdoor's intrusion could be via instillation code (software), such as 'Back Orifice' or through related backdoors that were done and left by the program's developers. Popular backdoor programs from the 90"s that used for mischief were Netbus Back, Orifice and Sub7. Examples to the latest popular backdoors are: Aimot, DsBot, Egg Drop, Hupigon, VanBot and Mo Sucker (Decloedt and Van Heerden, 2017).

Dai et al. (2019) carried out a black-box backdoor attack against LSTM-based text classification system with a random injection strategy in generating poisonous samples. Backdoor attack in CNN are the focus study points of view and improvements has been seen in term of attack and defense.

2.4 Backdoor Detection

There are many ways that make the computer gets infected by the backdoor attack. Backdoors attack can be bundled with shareware or other download software. It is not difficult for many types of the backdoor to pass the firewall of the system (Mudzingwa and Agrawal, 2014; Salimi and Arastouie, 2016). Therefore, most of these systems are provided with the second defense wall that is the backdoor detector, it is any technique or method that uses to protect the computer. The backdoor detection may or may not combine with the operating system. Some of the selected machine learning algorithm was put into consideration based on some available literature that uses them for detection purposes (Ahmed, 2017).

Moldovan, (2020) presented an improved classical version of the binary particle swarm optimization (BPSO) algorithm that brings in a specific type of particles called sensor particles. discussed the familiarization of that algorithm for data generated by sensors that monitors Daily Living Activities (DLAs) to find out the best position and features of the

monitoring sensors that gives the best classification output. A Machine learning method that joins the modified version of the algorithm was used to evaluate and validate their proposed approach. The method was tested and validated using Daily Life Activities (DaLiAi) dataset.

According to Decloedt (2017), the backdoor's detections have two inputs. First, the database (signature) or the knowledge of the backdoor is behavior. Second, the software that is under test. Generally, these detectors compare the backdoor's signature with the known patterns (database). This type called signature technique.

However, this technique cannot face a new backdoor's code (Mudzingwa and Agrawal, 2014; Salimi and Arastouie, 2016). Anomaly based detection uses its knowledge to check the normal behavior and detect the backdoor. This type includes a special rules set in order to decide, is it backdoor or not. However, this method cannot detect a lot of polymorphic viruses (Modi et al., 2013; Mudzingwa and Agrawal, 2014). The organization of backdoor detection, each technique can use one of three methods: dynamic, static, or hybrid.

In general, the backdoor detection techniques classify as follows:

a) Anomaly based Detection

i) Dynamic Anomaly

ii) Static Anomaly

iii) Hybrid Anomaly

b) Signature-based detection

i) Dynamic Signature

ii) Static Signature

iii) Hybrid Signature

c) Specification-based Detection

i) Dynamic Specification

ii) Static Specification

iii) Hybrid Specification

2.5 Machine Learning Algorithm

Machine Learning is a field of science that deals with the design of computer programs and systems that can learn rules from data, adapt to changes, and improve accuracy or performance with experience. Popularity of machine learning has the ability to perform two tasks. First the task that can be done by machines, second the task that can't be performed by humans. The Learning activity of machine learning makes it intelligent and also gives the system the ability of keeping up with changes of its environment. For this study, out of the many machine learning algorithms nine selected algorithms, Bayes Net, Bayesian LR, Naives Bayes, Naive Bayes, MultiLayer Perceptron, K-star, Stacking**,** Threshold Selection**,** Randomization filter Classifier and Zero R to study and identify the best classifier for detection of backdoor attack.

Multilayer Perceptron is a form of neural network in which the data is provided to the input layer, where it passes from one or more hidden lanes and results are made from the output layer. Such method is suitable for the Classification problems where class is specified in the input. Data is given in the form of tables Madhavi *et al.,* (2019).

2.6 Empirical Framework

Due to the exponential rise of information sharing on internet, this has resulted in challenging issues as it relates to the mining of data and machine learning concerning a backdoor attack. This has prompted for researches on comparative studies in terms of classification algorithms performance to accurately classify backdoor using a combination of performance metrics (Abu-nimeh *et al.*, 2017). It is, therefore, a matter of importance to determine algorithms that perform optimally for any chosen metrics to assist in proper classification of backdoor and legitimate site.

Dagar and Dahiya (2020) They came up with a binary particle swarm optimization (BPSO) based on edge detection method minimizing multi-objective fitness function, They formulated the function using weighted sum of five cost factors and all these cost factors that has to do with four methods of edge validation. The proposed approach was tested on 500 “BSD” images and the outcomes were compared with classical edge and computational intelligent methods (ACO, GA) using F-score performance parameter. Their proposed approach was in line with all images testing and outperform all classical edge detectors. ACO and GA having average F-score 0.2901 and has small standard deviation 0.0401.

Muduli *et al.* (2019) presented an optimized fuzzy logic-based fire monitoring system for wireless underground sensor network, to strengthen accuracy of deciding the mine fire prevention. But due to the large fuzzy rule set in the memory constraint sensor nodes, they used the binary particle swarm optimization (BPSO) algorithm for optimizing their proposed fuzzy system that abolishes redundant rules but keeps event detection accuracy of the monitoring system. BPSO algorithm and fuzzy logic toolbox inbuilt in the MATLAB was used in simulating the proposed system for the coal mining.

Saleh and Mohd (2019) proposed backdoor attack detection based on stepping stone detection approach. They achieved an actual detection rate of backdoor and ham of 98.7 %. Alex *et al.,* (2019) proposed detection of backdoor using machine learning algorithm that is Support Vector Machine (SVM). They achieved accuracy in detection of backdoor, but could not measure FP, FN, TP and other parameters. According to (Ahmed, 2018) proposed the detection of backdoor using KNN, classification decision, SVM, Naïve Baye, Ramdom Forest and J48 as machine learning algorithm. They achieved accuracy of 96.8% with the machine learning algorithms. But there is not robustness and applicability of the end classifier. Bryant *et al.,* (2018) detecting backdoor attacks on deep neural networks by activation clustering; the model achieved a recall of 95% however the accuracy was not recorded. (Te Juin *et al.* 2019), proposed Bypassing Backdoor Detection Algorithms in Deep Learning. They have showed that a sophisticated attacker is easily able to hide the signals of the backdoor images in the latent representation, rendering the defense ineffective. There was no record of any evaluation metrics. Huicong and Aspen (2017) proposed Low-cost Detection of Backdoor Malware, accuracy of 96% F1 score of 83% was achieved, however, there are still improvements to be made to the system.

2.7 Artificial Neural Network

(Isaac, Jantan, & Esther, 2018) presented Artificial neural network application issues, improvements, related performance and assessment method. The research separate across many applications of ANN methods in different fields which includes medicine, computing, agriculture, technology, engineering, environmental, science, climate, art, business and nanotechnology. The research found out that neural network models such as feedforward and feedback propagation artificial neural networks are better performing in its application to human problem. The proposed feedforward and feedback propagation

ANN models based on data analysis factors such as accuracy, scalability, processing speed, convergence and performance.

Samer, Jerjawi, & Abu-naser, (2018) used ANN in detecting whether someone is diabetic or not. Their aim was to reduce the error function in neural network training using a neural network model. After they carried out the ANN model training, the average error function of the neural network was equal to 0.01 and the accuracy of the prediction of maybe someone is diabetic or not was 87.3%.

Mohammed et al., (2020), they came up with an artificial neural network for predicting maybe lung cancer is found or not in the human body. Signs were used to know a person ssuffering from lung cancer; these signs were such as yellow fingers, coughing, chest pain, anxiety, fatigue, wheezing, shortness of breath, chronic disease and swallowing difficulty. They use those signs and other information of a sick person as input variables for the proposed ANN model. They carried out a training of the proposed model with using the lung cancer dataset in validating. The model was evaluated and tested with accuracy rate of 99.01%. (Salah, Altalla, Salah, & Abu-naser, 2018) designed and tested Artificial neural network (ANN) model that detect the rate of medical expenses using some number of factors that affect treatment expenses. They developed and trained a multi-layer perceptron topology based model using data on 5574 cases. They got a total score of 88% after the model was tested.

(Wu & Feng, 2017) did a review, summary of artificial neural network, its related theory, and the four main characteristics of ANN, which includes the non-linear, non-limitative, non-qualitative and non-convex and talked about its application in various fields. They also summarized the futures trends of ANN.

(Rad, 2018), presented a neural network model which was able to classify an unseen portable executable (PE) files as a benign or malicious, based on its loaded library function calls. The proposed model got an average accuracy of 97.8% with 97.6% precision and recall of 96.6% over a 4000 dataset size where 3,000 was malicious and 1,000 was benign PE files.

(Podder, Bharati, & Mondal, n.d.) They presented a systematic review on application of deep learning (DL) methods for cybersecurity with a brief description of DL techniques used in cybersecurity also involving deep belief network, recurrent neural networks, generation adversarial networks and other. They also showed the differences between DL and shallow learning. The study discuss on the feasibility of DL systems for classification and malicious attack detection, intrusion detection and other common attacks on cyberspace such as spam, network traffic and identifying file type. Their review shows high classification accuracy of 99.72% gotten by closed Boltzmann machine (RBM) when done on an edited dataset while long short-term memory (LSTM) got an accuracy of 99.80% for KDD cup 99 dataset.

(Sarker, 2021), did a comprehensive review from the aspect of neural networks and deep learning methods based on modern needs. They discuss the applicability of some techniques in different filed of cybersecurity issues like phishing, intrusion detection, backdoor detection, bonnets. They summarized and gave a future direction in the aspect of ANN.

2.8 Summary of Related Works

Table 2.1: Summary of Related Works

REFERENCES	PROBLEM	METHODOLOGY	ACHIEVEMENT	LIMITATIONS
Saleh and Mohd (2019)	To proposed backdoor attack detection based on stepping stone detection approach	stepping stone detection approach	They achieved an actual detection rate of backdoor and ham of 98.7 %. Could classify the traffic independently.	No record of false positive rate.
Alex *et al.*, (2019)		Machine Learning (Support Vector Machines (SVM))	Achieved a higher accuracy in detection backdoor attack.	Other evaluation metrics were not recorded
Ahmed (2017)	To detect backdoor using machine learning algorithm	Machine Learning KNN, Classification Tree, SVM, Naïve Bayes, Random Forest and J48).	They achieved accuracy of 96.8% with Classification Tree, SVM, Naïve Bayes, and J48 algorithms.	There is not robustness and applicability of the end classifier.
Bryant *et al.*, (2018)	To detect backdoor attack using some machine learning algorithm	Activation clustering	The model achieved a recall of 95% .	Accuracy was not recorded
Te Juin *et al.* 2019)	To detect backdoor attacks on deep neural networks	Deep Learning	They showed that a sophisticated attacker is easily able to hide the signals of the backdoor	There was no record of any evaluation metrics

Huicong Loi and Aspen Olmsted (2017)	Proposed Bypassing Backdoor Detection	Low - Cost	SVM which gives an accuracy 96% F score of 83%	
	Proposed Low-cost Detection of Backdoor Malware.			There is still improvements to be made to the system

CHAPTER THREE

3.0 RESEARCH METHODOLOGY

In this chapter, a dataset gotten from (NUSW-NB15) was used to realize the aim and objectives of this research. Figure 3.1 shows the methodology deployed to achieve each objective. It starts with investigation of machine learning models, feature selection using BPSO, development of the backdoor attack detection model and performance evaluation.

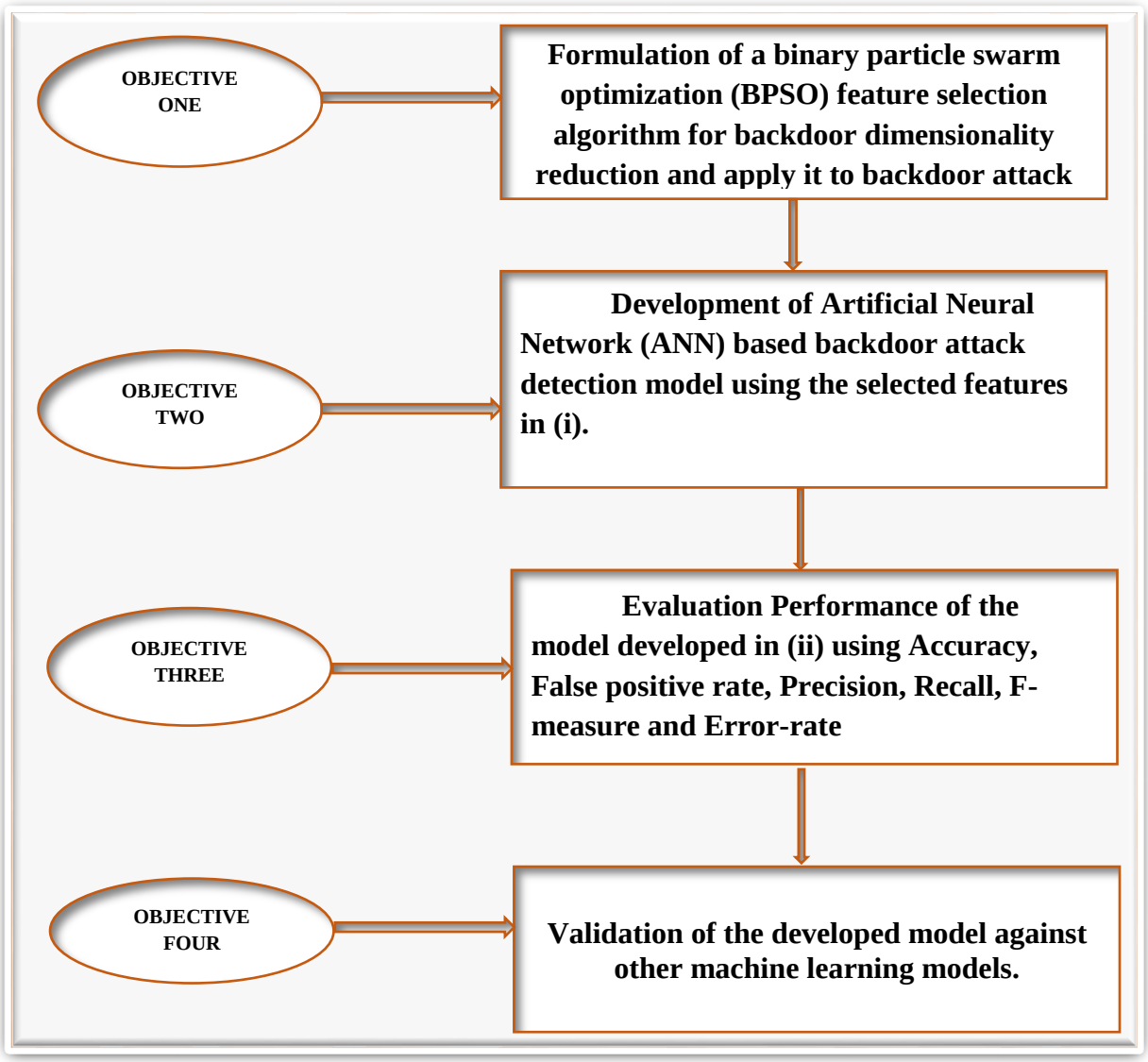

Figure 3.1: Research Methodology

3.1 Binary Particle Swarm Optimization (BPSO) Feature Selection

Relevant features were used to train the adopted model was selected using BPSO. This approach is a wrapper-based approach where MLP was used as the evaluator. There are three issues to consider while formulating the feature selection problem as an optimization problem. They are: the solution representation, and the fitness function.

3.1.1 Solution representation

Feature selection problem is a binary optimization problem where features are either selected or not. The solution are therefore represented as a vector with binary values of 1 and 0 representing a selected and rejected features respectively. Each solution is represented by a particle whose length is the number of features and values are 0 and 1 for features to be selected and those that should not be selected.

3.1.2 Fitness Function

In order to validate the performance of each particle and ascertain its fitness, mean absolute error (MAE) was computed as shown in Equation 3.7.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|A_i - P_i| \tag{3.7}$$

Where, N is the total number of samples, A_i is the actual class of sample i, P_i is the predicted class of sample i. Figure 3.2 shows the pseudocode of the BPSO algorithm for feature selection.

Pseudocode for BPSO

i. **Start**
ii. Set all BPSO parameters
iii. Initialize Particle positions randomly
iv. **While** *iter* < *Maximum_iter*
    a. **Evaluate** particle fitness
    b. **For** each particle P_i;
        i. **If** fitness of P_i < $Pbest_i$
            $Pbest_i = P_i$
        ii. **end if**
        iii. **If** fitness of $Pbest_i$ < $gbest_i$
            $gbest_i = Pbest_i$
        iv. **end if**
        v. **for** each dimension ***d,***
            update velocity

$$V_{id}(t+1) = w * V_{id}(t) + c_1 * r_1 * \left(P_{id} - x_{id}(t)\right) + c_2 * r_2 * \left(P_{gd} - x_{id}(t)\right) \quad (3.8)$$

            update particle position

$$\boldsymbol{x}(t+1) = \begin{cases} 1, & if\ rand < S(v(t+1)) \\ 0, & otherwise \end{cases} \quad (3.9)$$

            Where, S(v(t)) is a sigmoid function given as:

$$S\left(v(t)\right) = \frac{1}{1+e^{-v(t)}} \quad (3.10)$$

        vi. **end for**
    c. **end for**
v. **end while**
vi. **stop**

Figure 3.2: BPSO Feature selection pseudocode

3.2 Proposed BPSO-ANN Based Backdoor Attack Detection Model.

Figure 3.3 shows the proposed backdoor attack detection model. The process begins with collection of datasets, preprocess the datasets, design the BPSO-ANN model, train model and select appropriate features and evaluate model performance. The dataset collection, preprocessing, and feature selection have been described in the previous sections. Table 3.1 are the parameter settings of the BPSO and adopted ANN model for detection of backdoor attack.

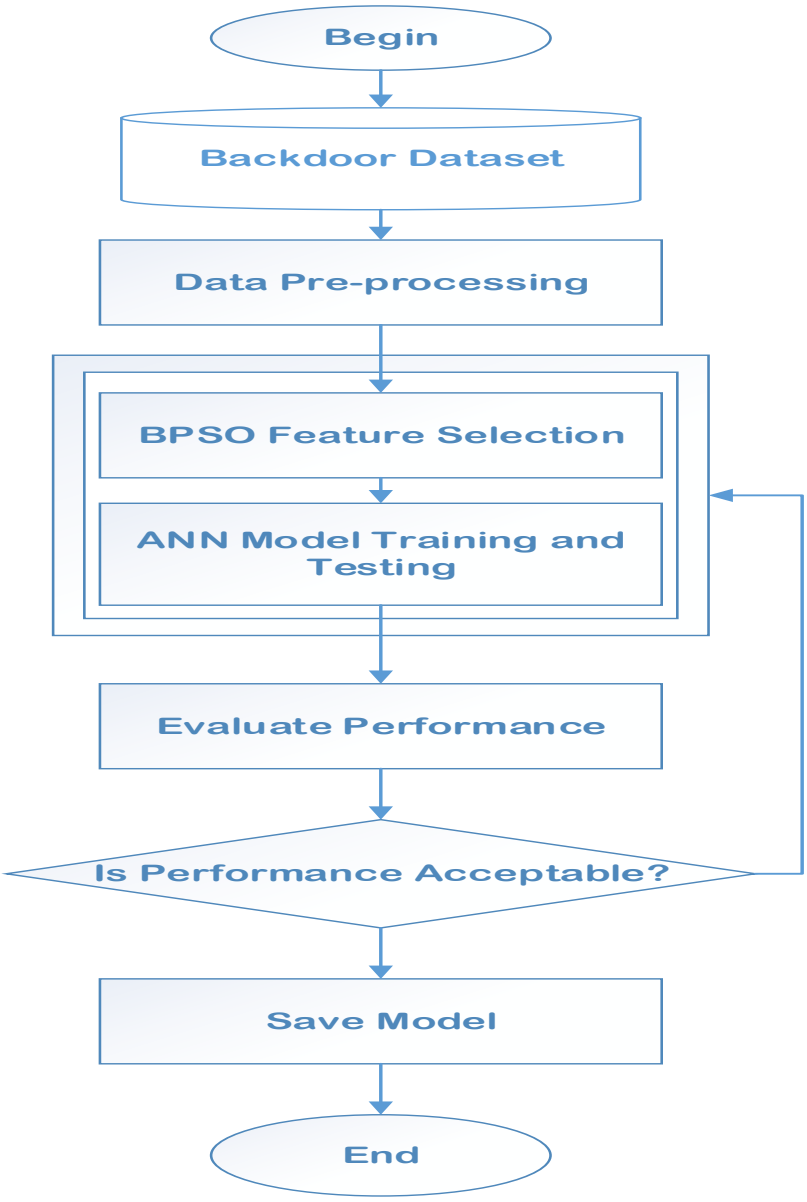

Figure3.3:BPSOANN Backdoor attack detection flowchart

Table 3.1: Parameter settings

Parameter	Value
Particle length	Number of features
Population size	10
Number of iteration	20
Maximum Inertia Weight (w_2)	2
Minimum Inertia Weight (w_1)	0
Acceleration constants (c1, c2)	(1, 2)
ANN Training function	'trainlm' (Levenberg-Marquardt backpropagation)
Hidden Layer size	30

The particle length or problem dimension is equal to the number of features, the maximum number of iterations is 20, the population size is 10, the minimum and maximum inertia weights are set to 0 and 2 respectively. A two layer MLP ANN was chosen with 30 hidden neurons with Levenberg-Marquardt backpropagation training function.

3.3 Artificial Neural Network Algorithm

The implementation steps of the adopted ANN used for training and testing of the proposed MLP-ANN

Pseudocode for MLP-ANN

---

i. ***Start***

ii. *Set all ANN parameters (number of layers, number of hidden nodes, activation function, training algorithms*

iii. *Randomly assign random weights to nodes*

iv. *Determine the activation rates of the input-hidden nodes*

v. *Determine the activation rates of the hidden-output nodes*

vi. ***Calculate*** *the error rate of the output nodes and re-calibrate hidden-output connections*

vii. *Recalibrate the weights between the hidden and input nodes using the weights and error from the output node*

viii. *Repeat the process until the convergence criteria is met*

ix. *Return the score of the output node using the final node connections and weights*

x. ***Stop***

Figure 3.4: ANN algorithm

3.4 Model Training and Testing

Figure 3.5 show the steps used in training and testing of the model.

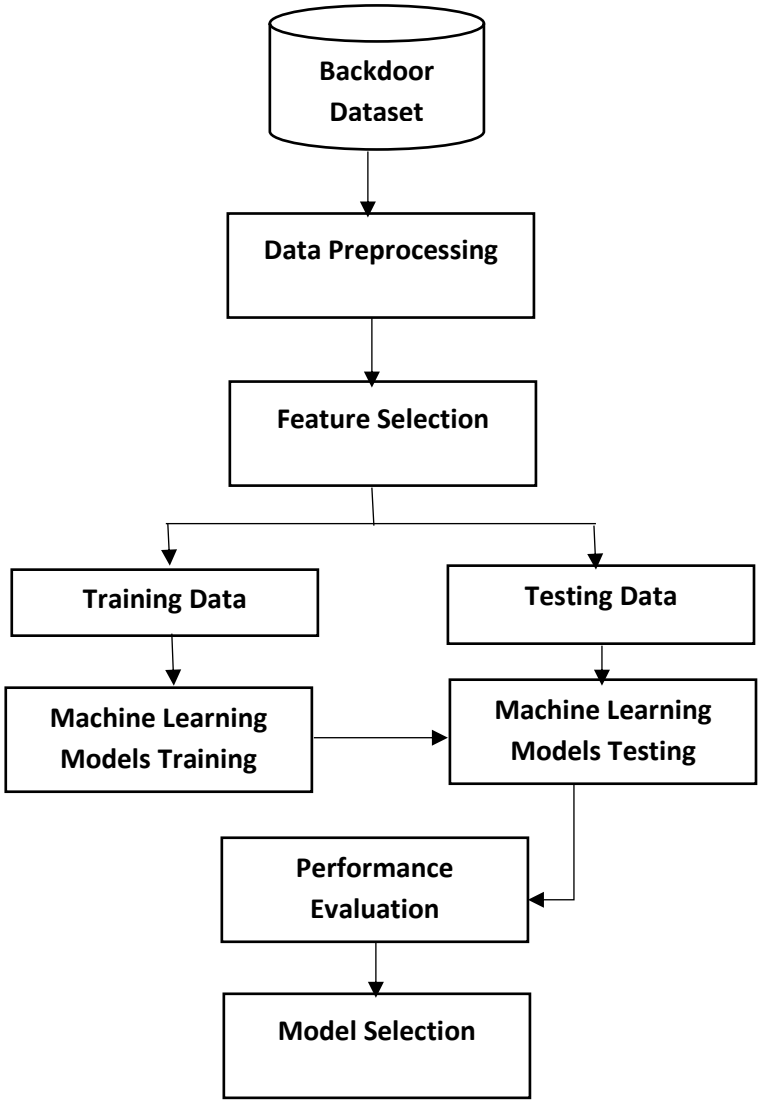

Figure 3.5: Machine Learning Model Training and Testing

This figure 3.5 shows the machine learning model training and testing pipeline. From literature, most of the proposed techniques do not apply feature selection algorithms for backdoor detection. Hence BPSO was applied to improve the performance of the backdoor detection.

3.4.1 Dataset collection and preprocessing

Backdoor dataset was collected from (NUSW-NB15) dataset which is made of backdoor attack. UNSW-NB15 data set was generated by IXIA perfect storm toll in the Cyber range lab of Australian Centre for Cyber Security (ACCS). This dataset has been created after the application of 12 algorithms and tools. TCP dump tool was used to generate 100 GB of traffic data, which contains collection of normal and abnormal activities. The downloaded dataset is comma-separated values (CVS) format. Attack and normal instances are combined and labeled with “benign” for normal traffic and backdoor for attack traffic, the dataset is made of 49 attributes and 4330 instances. The collected dataset was preprocessed by removing the instances that are non-numeric in nature, also the data was manually cleaned up by removing blank spaces. Due to large number of features, the feature selection algorithm is applied to select the most relevant and promising features. Feature selection is very necessary in machine learning algorithm, because it helps in removing useless features from the dataset. For this study, the feature selection process was done by dropping the features with least importance using ranker search and correlation attribute evaluator in WEKA environment.

Table 3.2: Sample Datasets and features

Features Samples	sport	dsport	Sintpkt	Dintpkt	tcprtt	synack	Label
1	25981	111	1.273667	1.108	0	0	0
2	27347	22309	0.460667	0.218333	0	0	0
3	61652	80	115.7807	74.90723	0.000677	0.000531	0
4	44623	80	83.67115	63.97647	0.000637	0.000491	0
5	42576	28305	0.333447	0.314633	0.000745	0.000607	0
6	41852	80	99.15264	64.15276	0.000654	0.000492	0
7	33559	53	0.011	0.006	0	0	0
8	23314	143	0.277909	0.267688	0.000653	0.000517	0
9	15524	45417	0.461	0.233	0	0	0
10	39077	53	0.007	0.002	0	0	0
11	45605	42279	12.9073	12.01069	0.000658	0.00052	0

In order to validate the developed BPSO-ANN model, a 70/30% train test split option was applied on other machine learning classification algorithms in the experiment using the complete dataset. The following nine classifiers were used; Bayes Net, Bayesian LR, NaiveBayes, SVM, Kstar, Stacking, Threshold Selection, Randomizable Filter and Zero R.

3.4.2 Performance Evaluation

The performance metrics used for measuring the performances of the selected classifiers are:

i. **Accuracy**

Accuracy is the proportion of all predictions that are correctly identified as “Attack record” and “Normal record”. Accuracy is the most intuitive performance measure of an

intrusion detection system. It directly reflects the superiority of the system. Accuracy is defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

ii. Precision

Precision is the proportion of all predicted attack records that are actually attack records. Precision is defined as follows:

$$\Pr ecision = \frac{TP}{TP + FP} \tag{3.2}$$

iii. Recall

Recall is a measure of coverage and can indicate the ratio of the actual attack records that are

correctly identified. Recall is also considered as the detection rate. With certain accuracy, the recall rate of the classifier is required to be as high as possible. The recall is defined as follows:

$$TPR = \frac{TP}{TP + FN} \tag{3.3}$$

iv. False Positive Rate (FPR)

FPR generally indicates the probability that normal records are incorrectly predicted as attack records. The FPR affects the performance of the intrusion detection system. The FPR is defined as: follows:

$$False \;\; Positive\, Rate = \frac{FP}{TN - FP} \tag{3.4}$$

v. F1-score

F1-score is the harmonic mean of precision and recall. In other words, it can be interpreted asa weighted average of the precision and recall. The F1-score takes false positives and false negatives into account. The F1-score is defined as follows:

$$\text{F –measure} = \frac{2TP}{2TP + FP + FN} \tag{3.5}$$

vi. Error Rate (ER)

Error Rate (ER) This refers to how often is the system wrong.

$$ErrorRate = \frac{FP + FN}{TP + FP + TN + FN} \tag{3.6}$$

The model with the best performance will be selected and used for the feature selection in the next stage.

CHAPTER FOUR

4.0 RESULTS AND DISCUSSION

4.1 BPSO-ANN Feature Selection Result

The features selected by the BPSO-ANN are shown in Table 4.1. Eleven features selected is indicated by √ symbol. The selected features include; dstip, dsport, dbytes, dttl, sloss, sload, dloss, dload, dmeansz, sintpkt, and dintpkt. The selected features were used to train the multi-layer perceptron (MLP) Artificial Neural Network (ANN) for the detection of backdoor attacks.

Table 4.1: BPSO-ANN and ANN Features Selected

S/N	Features	BPSO-ANN Selected Features
1	Sport	
2	dstip	√
3	dsport	√
4	sbytes	
5	dbytes	√
6	Dttl	√
7	Sloss	√
8	Dloss	√
9	Sload	√
10	Dload	√
11	Spkts	
12	Dpkts	
13	Dmeansz	√
14	Sintpkt	√
15	Dintpkt	√
16	tcprtt	
17	synack	

Comparing the Performance of MLPANN before and after applying BPSO for Feature Selection

Table 4.2 show the results for MLPANN and after applying BPSO, from the selected eleven features out of the seventeen features.

Table 4.2: Results for MLPANN and after applying BPSO. From the selected 11 features out of the 17 features

Approach	Accuracy	False Positive Rate	Precision	Recall	F-Measure
BPSO+MLPANN	99.97	0.031	0.99	0.99	0.99
MLPANN	99.85	0.053	0.97	0.971	0.971

4.2 BPSO Convergence Result

Figure 4.1 shows the convergence curve of the BPSO algorithm. It shows the maximum iteration and global best fitness obtained for each iteration. From the curve, the minimum or global best value obtained is 0.3e-5. A maximum of 20 iterations was set and the algorithm starts converging from 2.2e-5 from the first iteration to 0.3e-5 from the 19^{th} iteration. This result shows the fastness in convergence by the BPSO algorithm in searching for the global solution.

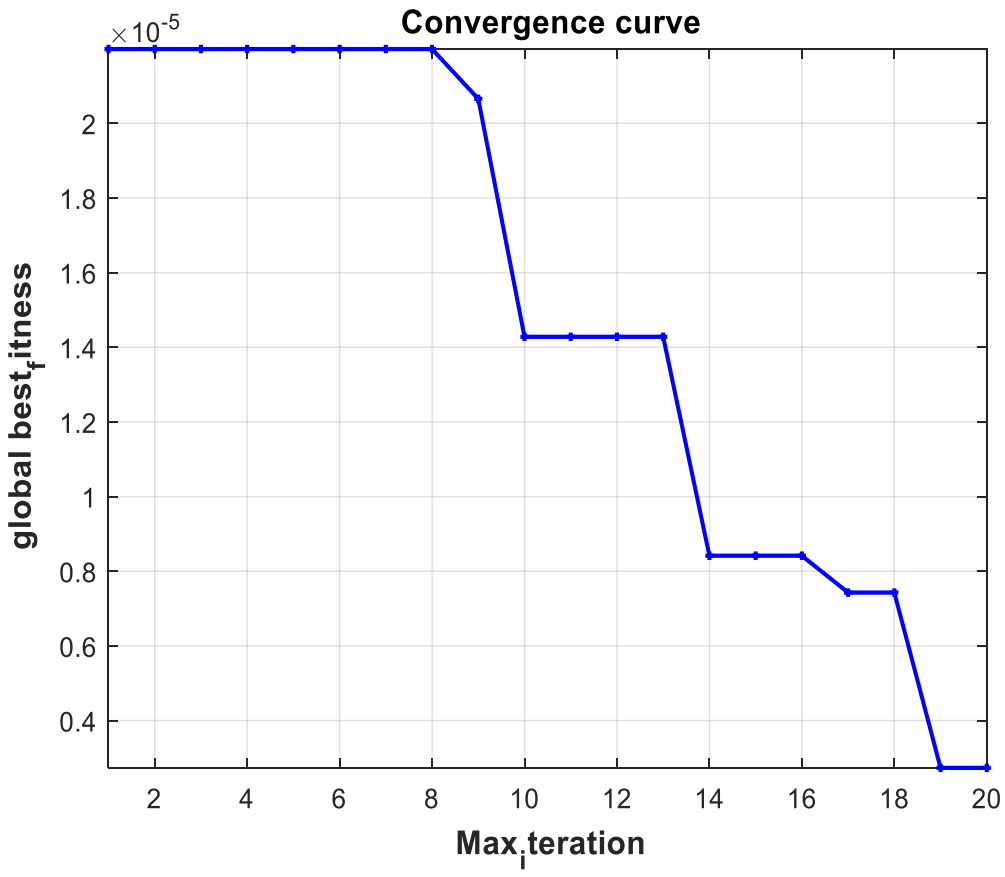

Figure 4.1: BPSO Convergence Curve

4.2.1 BPSO-ANN detection results

Figure 4.2 shows the confusion matrix of the test performance of the proposed BPSO-ANN classifier. The results indicated that no misclassification was recorded. From figure 4.2, 389 signifying 44.9% of entire test set were correctly classified as normal while 447 samples signifying 55.1% of test samples were correctly classified as attack. This means that all samples were correctly classified as attack or normal samples.

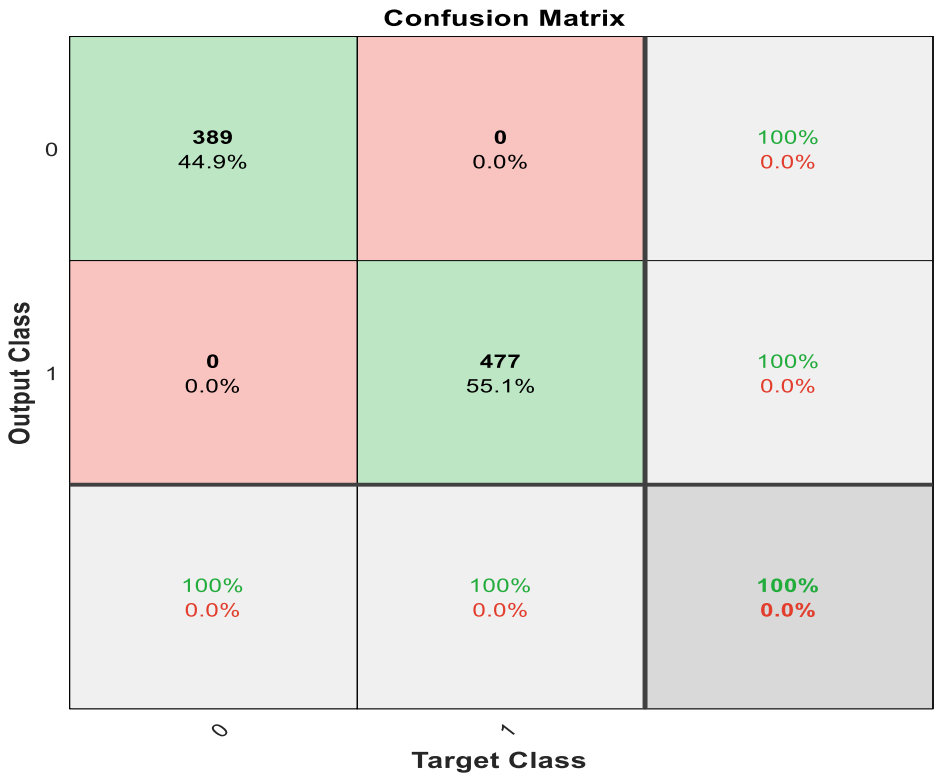

Figure 4.2: Test Confusion Matrix

Table 4.3 summarizes the performance of the model in terms accuracy, sensitivity, specificity and precision. The results show a 99.97% accuracy, precision of 0.99, recall of 0.99, f-measure of 0.99 and false positive rate of 0.031

Table 4.3: performance evaluation of BPSO-ANN

Accuracy	False Positive Rate	Precision	Recall	F-Measure
99.97	0.031	0.99	0.99	0.99

4.3 Performance Validation Result

Table 4.2 shows the performance validation of the developed BPSO-ANN model against other machine learning models using Accuracy, False positive rate, Precision, Recall, and F-measure. From the results analysis, it's evidently clear that the MLP classifier performed better than other models and therefore adopted for the second phase of this research work.

Similarly, when all 17 features were used to train the MLP-ANN model, 99.97% accuracy, sensitivity, specificity and precision were obtained. Therefore, the proposed BPSO-ANN model with less number of features and highest performance was adopted for detection of backdoor attacks.

Table 4.4: Performance validation

Classifiers	**Num of Features**	**Accuracy**	**False Positive Rate**	**Precision**	**Recall**	**F-Measure**
BPSO- ANN	**11**	**99.97**	**0.031**	**0.99**	**0.99**	**0.99**
Bayes Net	17	99.28	0.007	0.993	0.993	0.99
Bayesian LR	17	88.68	0.098	0.907	0.887	0.886
NaivesBaye	17	93.16	0.077	0.936	0.932	0.931
Lib SVM	17	85.81	0.122	1	0.858	0.857
K-star	17	98.75	0.011	0.892	0.988	0.975
Stacking	17	53.81	0.538	0	0.538	0
Threshold Selection	17	99.77	0.042	0.997	0.997	0.997
Ramdomasable Filter	17	96.33	0.037	0.963	0.963	0.963
Zero R	17	53.81	0.338	0	0.538	0

The discussion of the results are as follows:

i. Accuracy

The highest accuracy of 99.97% on BPSO-ANN was obtained and 53.27% with stacking and Zero R as shown in Fig. 4.3.

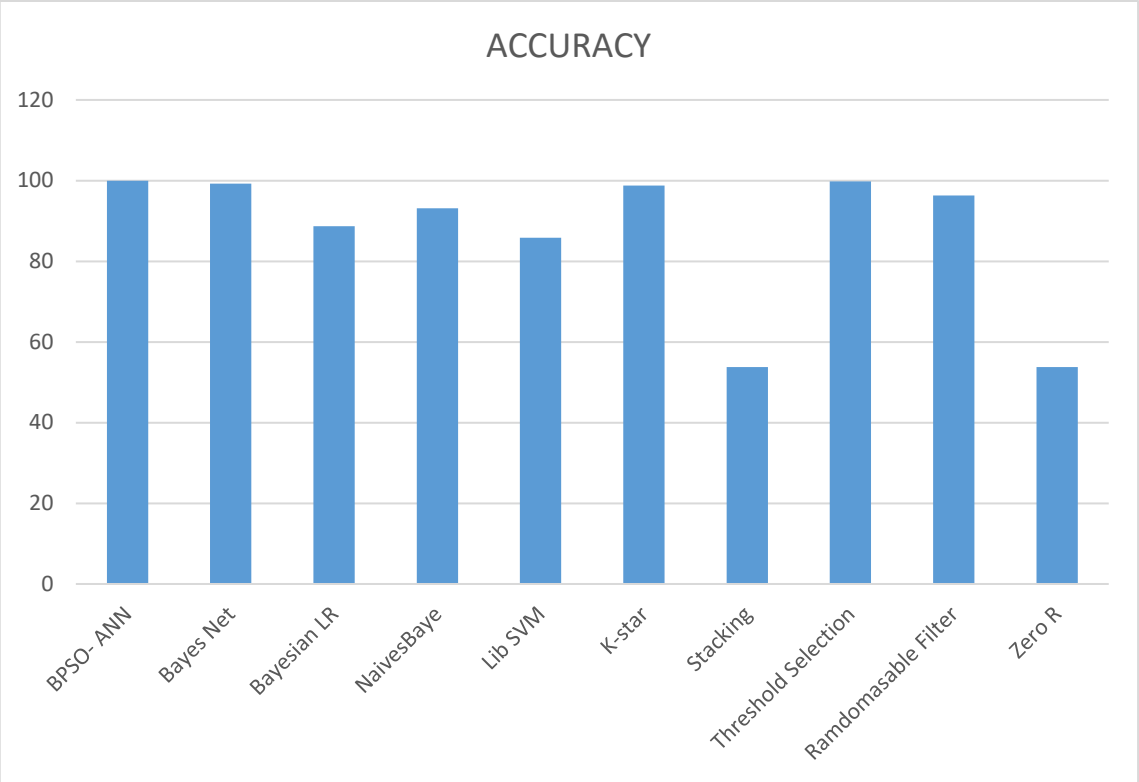

Figure 4.3 Comparison of Accuracy

ii. False Positive Rate (FPR)

The FPR is the number of legitimate sites incorrectly classified as backdoor. The BPSO-ANN obtained the lowest FPR of for 10 folds split. Stacking and Zero R with 70/30% split test indicate the False Positive Rate of 0.533 with stacking and Zero R which is the highest FPR as shown in Figure Fig. 4.4.

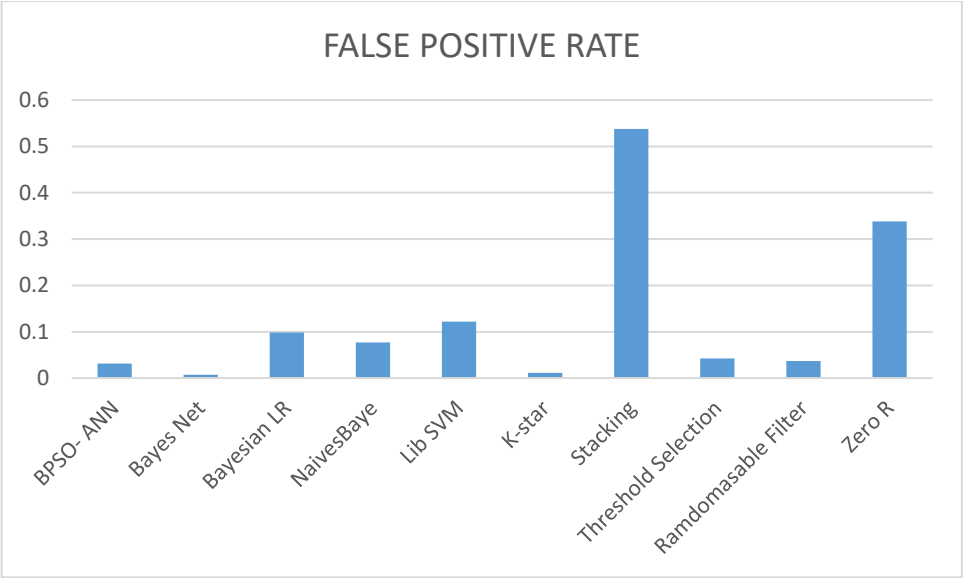

Figure 4.4 Comparison of False Positive Rate

iii. Precision

Precision indicates the fraction of important recollected instances. BPSO-ANN with a precision of 1.00 under 10 fold cross-validation outperformed other classification algorithms with little variation compared to stacking and Zero R with 0.00 for 70/30% split test indicate the precision of 0.99 on BPSO-ANN was obtained and 0% with stacking and Zero R as shown in Figure 4.5.

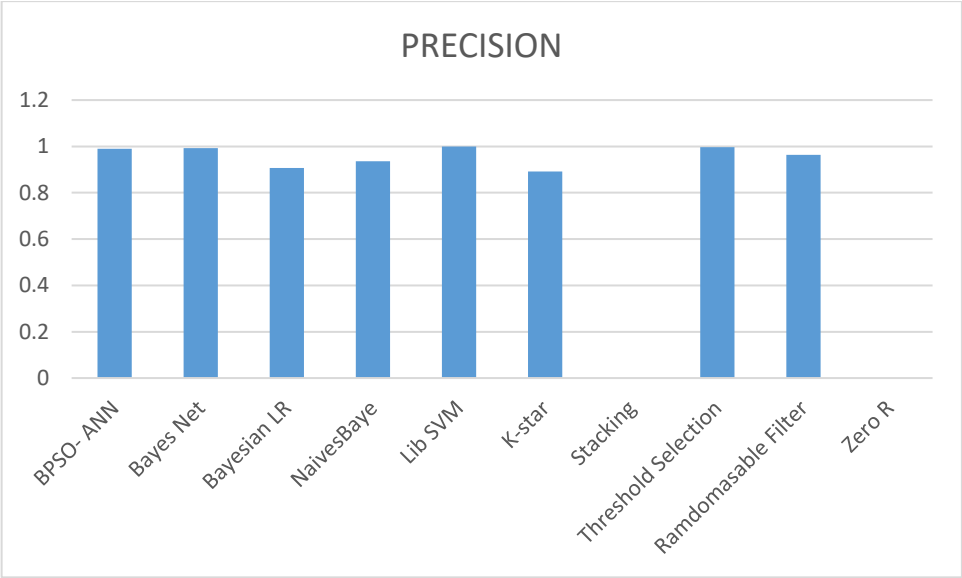

Figure 4.5 Comparison of Precision

iv. Recall

Recall represents the relevant instances which are recalled. BPSO-ANN attained the best recall rate of 0.99 under 10-fold cross-validation test option outperforming other classification algorithms. With 70/30% split test indicate the precision of 0.99 on Multilayer perceptual was obtained and 0.538 with stacking and Zero R as shown in Figure 4.

v. F-Measure

A high F-Measure is needed since both precision and recall are needed to achieve a high score and BPSO-ANN has the highest F-Measure of 0.99 and 0.00 for ZeroR as presented in Figure 4.6 through the application of a 10 fold cross-validation test option. With 70/30% split test indicate the precision of 0.99 on BPSO-ANN was obtained and 0.00 with stacking and Zero R.

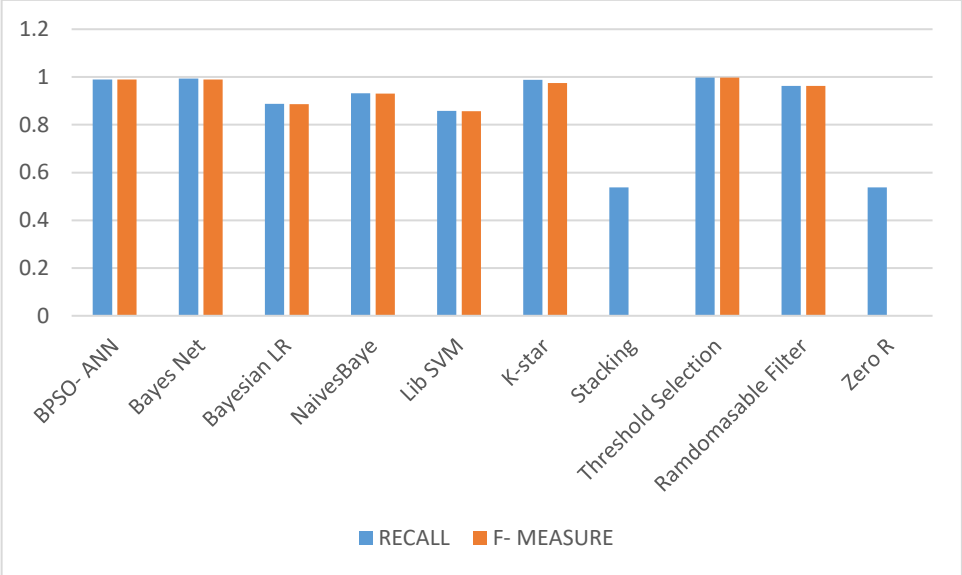

Figure 4.6 Comparison of Recall and F-Measure

Table 4.5: Performance Validation

Reference	Approach	Accuracy %
Proposed Model	**BPSO-ANN**	**99.97**
Saleh and Mohd (2019)	stepping stone detection approach	98.7
Ahmed (2017)	Classification Tree, SVM, Naïve Bayes, Random Forest and J48).	96.8
Huicong Loi & Aspen Olmsted (2017	Low - Cost	96

The proposed model performs better in terms of accuracy when compared with existing models.

CHAPTER FIVE

5.0 CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The study that was simulated as a result of the exponential rate of backdoor attack across Internet platform and also as a result of knowledge derived from the reviews of literature available on classification algorithms that are yet to be compared in terms of their performance metric on backdoor attack datasets.

Machine learning classification algorithms used in performing experiment on backdoor dataset involving two test options, Multilayer perceptron outperform other machine learning classification algorithms proving its efficiency in performance based. The indication from the result obtained from the experiment reveals that BPSO-ANN with 99.97% accuracies, 0.031false positive rate was excellent for backdoor detection classification, which is effective than other commonly recognized classification algorithms such as stacking, and Zero R with 53.81%, 53.81% accuracies, 0.538 and 0.338 false positive rate respectively.

When BPSO was applied for feature selection, 11 features were selected and a 99.97% accuracy, false positive rate, precision, recall and F-measure were recorded for the BPSO-ANN model. Therefore, the proposed model was adopted and recommended for use in backdoor attack detection.

5.2 Recommendations

An anti-backdoor application developer can implement the proposed BPSO and machine learning classification algorithm that was discovered to be the best in this study to enhance the feature of backdoor attach detection and classification, thereby reducing high false-positive rate associated with some methodology employed in backdoor attack detection furthermore, helping in filtering backdoor attack.

For future work, more optimization algorithms can be investigated to ascertain their effect in selecting features and building a more efficient backdoor attack detector.

5.3 Contribution to Knowledge

The contribution of this research is hereby outlined;

i- An optimized MLPANN classifier that was able to achieve a better performance in terms of relevant metric used in related literatures

ii- The proposed model (BPSO-ANN) performs better in terms of accuracy when compared with existing models.

iii- The developed model shows Accuracy of 99.97%.

REFERENCES

Abu-nimeh S., D. Nappa, X. Wang, and S. Nair, (2017) "A Comparison of Machine Learning Techniques for backdoor Detection," in *In Proceedings of the anti-backdoor working groups 2nd annual eCrime researchers summit*, 60–69.

Ahmed D. A., (2017). Detection Using Machine Learning Algorithm. *International Journal of Computer Science, Engineering and Information Technology*,7,(6), 1 - 8. DOI : 10.5121/ijcseit.2017.7601.

Alex, G., Praveena D, & Ereethi H., (2019). Detection of backdoor using machine learning algorithm that is Support Vector Machine (SVM). *International Journal of Innovative Research in Science, Engineering and Technology*. 8, (3), 1751-1754. DOI:10.15680/IJIRSET.2019.0802010.

Arican, M., & Polat, K. (2020). *Binary particle swarm optimization (BPSO) based channel selection in the EEG signals and its application to speller systems*. 27–37. https://doi.org/10.33969/AIS.2020.21003

Bal, R., & Sharma, S. (2016). Review on Meta Classification Algorithms using WEKA, *35*(1), 38–47,

Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, & Biplav Srivastava (2018). Detecting Backdoor Attacks on Deep Neural Networks by Activation Clustering. *Paper presented at the Computational and Information Sciences International Conference on.*2(7), 6 – 12.

Chang H. K.,Kabanga E. K., &Sinjae K. (2018). Efficiency-based comparison on Malware Detection Techniques. *International Conference on Platform Technology and Service*, 6(7), 1 – 7.

Choi, W. S., & Choi, S. G. (2013). An enhanced method for mitigation of network traffic using TCP signalling control. Paper presented at the Advanced Communication Technology.

Chris, Wysopal & Chris Eng (2016). Detection of application backdoors. *The Journal of Machine Learning Research*, 9(8), 949-979.

Dada, E. G, Joseph S. B., Yakubu J. H, & Abdulkadir H. A, (2019). Performance Evaluation of Machine Learning Algorithms for Detection and Prevention of Malware Attacks. *Journal of Computer Engineering.* 21, (3), 18-27.

Dagar, N. S., & Dahiya, P. K. (2020). ScienceDirect ScienceDirect Edge Detection Technique using Binary Particle Swarm Edge Detection Technique using Binary Particle Swarm Optimization Optimization. *Procedia Computer Science*, *167*(2019), 1421–1436. https://doi.org/10.1016/j.procs.2020.03.353

Dai, J., Chen, C., & Li, Y. (2019). A Backdoor Attack Against LSTM-Based Text Classification Systems. *IEEE Access*, *7*, 138872–138878. https://doi.org/10.1109/ACCESS.2019.2941376

Decloedt, H. E., & Van Heerden, R. P. (2017). Rootkits, Trojans, backdoors and new developments. CSIR. Defence Peace Safety and Security.

http://hdl.handle.net/10204/4689

Elham Salimi &Narges Arastouie (2016). Backdoor Detection System Using Artificial Neural Network and Genetic Algorithm. *International Conference on Computational and Information Sciences*. 817 -820.

Huicong Loi & Aspen Olmsted (2017). Low-cost Detection of Backdoor Malware. Learning Techniques. *International Journal of Innovative Research in Science, Engineering and Technology*. 8, (3), 1751- 1754. DOI:10.15680/IJIRSET.2019.0802010.

Isaac, O., Jantan, A., & Esther, A. (2018). State-of-the-art in arti fi cial neural network applications : A survey. *Heliyon*, (October), e00938. https://doi.org/10.1016/j.heliyon.2018.e00938

Ji, B. A. I., Lu, X., Sun, G., Li, J., & Xiao, Y. (2020). *Bio-Inspired Feature Selection : An Improved Binary Particle Swarm Optimization Approach*. *8*, 85989–86002. https://doi.org/10.1109/ACCESS.2020.2992752

Khalid A. & Mohd N. O. (2016). Backdoor Attack Detection Based on Stepping Stone Detection Approach. *IEEE International Conference on Sensors and Nanotechnology*, 2, (5), 88-92.

Kwon, H. (2020). *Detecting Backdoor Attacks via Class Difference in Deep Neural Networks*. 191049–191056. https://doi.org/10.1109/ACCESS.2020.3032411

Maarof, M. A., & Osman, A. H. (2012). Malware Detection Based on Hybrid Signature Behaviour Application Programming Interface Call Graph. *American Journal of Applied Sciences*, 9(6) 23 -28.

Madhavi Dhingra, S C Jain, Rakesh Singh Jadon (2019). Malicious Intrusion Detection Using Machine Learning Schemes. *International Journal of Engineering and Advanced Technology*, 8 (6), 2249 – 8958.

Microsoft (2012). Microsoft Security Intelligence Report "Worldwide Threat Assessment" (Vol. 13): Technical Report.

Modi, C., Patel, D., Borisaniya, B., Patel, H., Patel, A., & Rajarajan, M. (2013). A survey of intrusion detection techniques in cloud. *Journal of Network and Computer Applications*. 5(7), 3 -8.

Mohammed, O., Kweik, A., Atta, M., Hamid, A., Sheqlieh, S. O., Abu-nasser, B. S., & Abu-, S. S. (2020). *Artificial Neural Network for Lung Cancer Detection*. *4*(11), 1–7.

Moldovan, D. (2020). *applied sciences Adapted Binary Particle Swarm Optimization for Efficient Features Selection in the Case of Imbalanced Sensor Data*. https://doi.org/10.3390/app10041496

Muduli, L., Mishra, D. P., Jana, P. K., & Member, S. (2019). Optimized Fuzzy Logic-Based Fire Monitoring in Underground Coal Mines : Binary Particle Swarm Optimization Approach. *IEEE Systems Journal*, *PP*, 1–8. https://doi.org/10.1109/JSYST.2019.2939235

Mudzingwa, D., & Agrawal, R. (2014). A study of methodologies used in intrusion detection and prevention systems. *Paper presented at the Southeastcon, Proceedings of IEEE.*

Podder, P., Bharati, S., & Mondal, M. R. H. (n.d.). *Artificial Neural Network for Cybersecurity : A Comprehensive Review*.

Prasad, M. S., Babu, A. V., & Rao, M. K. B. (2013). An Intrusion Detection System Architecture Based on Neural Networks and Genetic Algorithms. *International Journal of Computer Science and Management Research.*

Rad, B. B. (2018). Malware classification and detection using artificial neural network. *Journal of Engineering Science and Technology*, *13*, 14-23

Saha, A., Subramanya, A., & Pirsiavash, H. (2020, April). Hidden trigger backdoor attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, No. 07, pp. 11957-11965).

Salah, M., Altalla, K., Salah, A., & Abu-naser, S. S. (2018). *Predicting Medical Expenses Using Artificial Neural Network*. *2*(10), 11–17.

Salimi, E., & Arastouie, N. (2011, October). Backdoor detection system using artificial neural network and genetic algorithm. In *2011 International Conference on Computational and Information Sciences* (pp. 817-820). IEEE.

Samer, N., Jerjawi, E., & Abu-naser, S. S. (2018). *Diabetes Prediction Using Artificial Neural Network*. *121*, 55–64.

Sarker, I. H. (2021). Deep Cybersecurity : A Comprehensive Overview from Neural Network and Deep Learning Perspective. *SN Computer Science*, *2*(3), 1–16. https://doi.org/10.1007/s42979-021-00535-6

Sushil k.K.,(2015). Analysis of WEKA Data Mining Algorithm REPTree, Simple Cartand RandomTree forClassification of Indian News, *International Journal of Innovative Science, Engineering & Technology*, 2(2), 438-446. Thompson, Ken, "Reflections on Trusting Trust", *Communication of the ACM* Vol. 27, No. 8, http://www.acm.org/classics/sep95/, Sep.2015.

Te Juin Lester Tan & Reza Shokri (2019). Bypassing Backdoor Detection Algorithms in Deep Learning. *International Journal of Engineering Research & Technology (IJERT),* 8 (07),135 – 139.

Thomas, S., Francillon, A., Thomas, S., Francillon, A., Definition, B., & Proceedings, D. (2018). *Backdoors : Definition , Deniability and Detection To cite this version : HAL Id : hal-01889981 Backdoors : Definition , Deniability and Detection*. (Raid).

Wu, Y., & Feng, J. (2017). Development and Application of Artificial Neural Network. *Wireless Personal Communications*. https://doi.org/10.1007/s11277-017-5224-x

Yoan, L. , Julio, M, & Ireimis, L. (2016). Study of the Performance of the K* Algorithm in International Databases. *International Journal of Engineering and Advanced Technology*, 12, (23), 51-56.