symmetry

MDPI

Article

A Feature Selection Model for Network Intrusion Detection System Based on PSO, GWO, FFA and GA Algorithms

Omar Almomani

Computer Network and Information Systems Department, The World Islamic Sciences and Education University, Amman 11947, Jordan; Omar.almomani@wise.edu.jo

Received: 3 June 2020; Accepted: 19 June 2020; Published: 23 June 2020

Abstract: The network intrusion detection system (NIDS) aims to identify virulent action in a network. It aims to do that through investigating the traffic network behavior. The approaches of data mining and machine learning (ML) are extensively used in the NIDS to discover anomalies. Regarding feature selection, it plays a significant role in improving the performance of NIDSs. That is because anomaly detection employs a great number of features that require much time. Therefore, the feature selection approach affects the time needed to investigate the traffic behavior and improve the accuracy level. The researcher of the present study aimed to propose a feature selection model for NIDSs. This model is based on the particle swarm optimization (PSO), grey wolf optimizer (GWO), firefly optimization (FFA) and genetic algorithm (GA). The proposed model aims at improving the performance of NIDSs. The proposed model deploys wrapper-based methods with the GA, PSO, GWO and FFA algorithms for selecting features using Anaconda Python Open Source, and deploys filtering-based methods for the mutual information (MI) of the GA, PSO, GWO and FFA algorithms that produced 13 sets of rules. The features derived from the proposed model are evaluated based on the support vector machine (SVM) and J48 ML classifiers and the UNSW-NB15 dataset. Based on the experiment, Rule 13 (R13) reduces the features into 30 features. Rule 12 (R12) reduces the features into 13 features. Rule 13 and Rule 12 offer the best results in terms of F-measure, accuracy and sensitivity. The genetic algorithm (GA) shows good results in terms of True Positive Rate (TPR) and False Negative Rate (FNR). As for Rules 11, 9 and 8, they show good results in terms of False Positive Rate (FPR), while PSO shows good results in terms of precision and True Negative Rate (TNR). It was found that the intrusion detection system with fewer features will increase accuracy. The proposed feature selection model for NIDS is rule-based pattern recognition to discover computer network attack which is in the scope of Symmetry journal.

Keywords: network intrusion detection system (NIDS); network security; feature selection; particle swarm optimization (PSO); grey wolf optimizer (GWO); firefly optimization algorithm (FFA); genetic algorithm (GA); UNSW-NB15; J48; SVM

1. Introduction

Vinchurkar and A. Reshamwala [1] defined an intrusion detection system (IDS) as "an active process or device that analyzes system and network activity for unauthorized and nasty activity." There are three types of IDS. These types are [2]: host-based IDS (HIDS), network intrusion detection system (NIDS) and hybrid-based IDS (HISD). The HIDS aims at tracking the internal activities of the computer system. The NIDS aims at tracking the network traffic logs dynamically in real time. It aims at doing that for identifying any potential intrusion into the network. It aims at doing that through employing the correct detection algorithms. Regarding the detection mechanisms that are

based on an IDS, they are classified into detection of misuse, detection of anomalies and hybrid IDS [3]. The mechanism of misuse detection is a group of predefined signatures or rules that aim at detecting known attacks. The mechanism of anomalies detection performs an activity that is normal for detecting unknown attacks. It does that through testing whether the device state is normal or not. Figure 1 shows the IDS classification of anomaly detection. Hybrid IDS detects attacks of known and unknown activity. This paper focuses on the NIDS. The NIDS detects attacks through using the entire network traffic feature. For detecting attacks, not all the features are required. A lower number of features can minimize the time needed for detection and increase the detection rate. In addition, feature selection has many benefits that are for the favor of the learning algorithms. For instance, it prevents over-fitting, it prevents noise resistance and it improves the predictive performance.

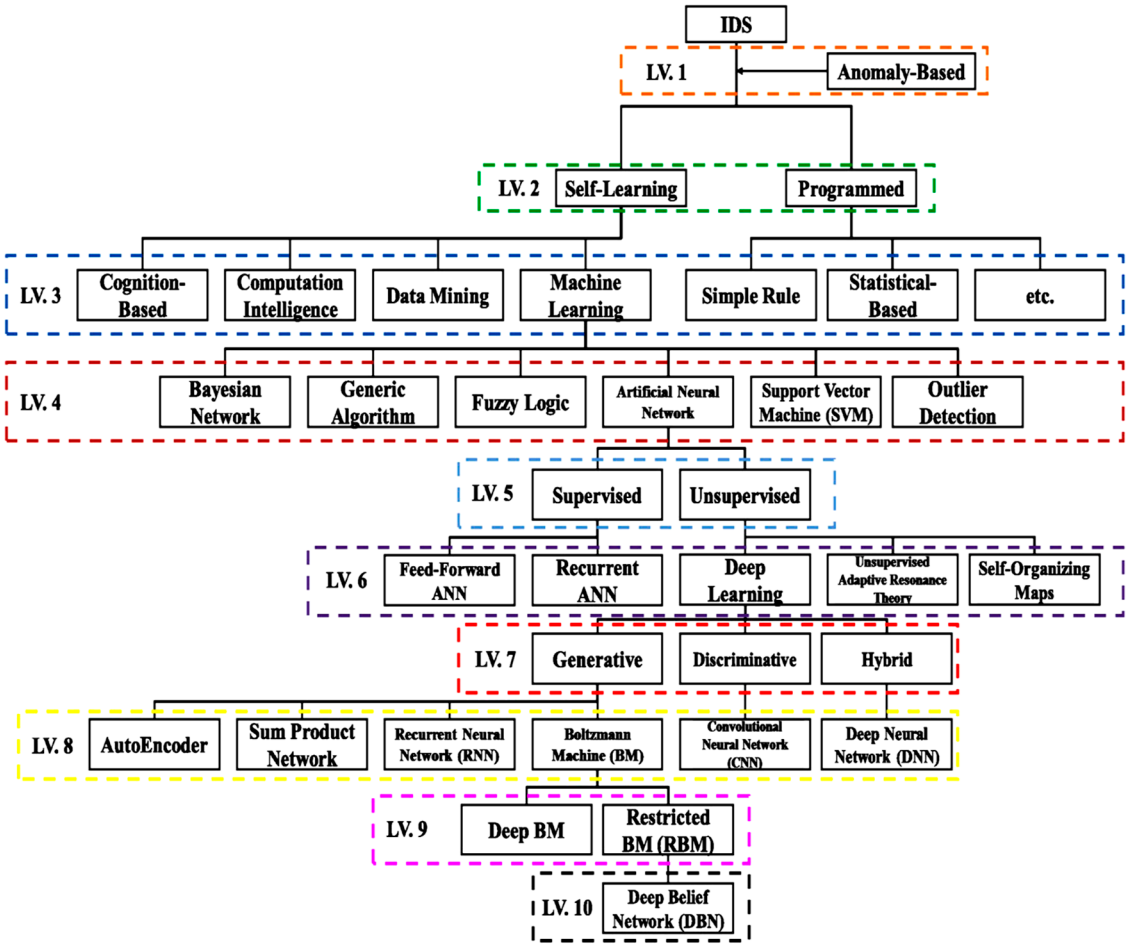

Figure 1. Classification of anomaly detection [4].

Feature selection is the technique for selecting a subset of relevant features to be used for model construction. It aims to enhance the data quality. The subset of features is called S in Equation (1) below:

$$S = \{s_1, s_2, s_3, s_4, \ldots.., s_{2n-1}\}. \quad (1)$$

Feature selection has been commonly used in many areas. For instance, it is used in IDSs. There are three techniques for selecting features [5–7]: wrapper [8], filter [9] and embedded techniques [10]. Through the embedded technique, the feature selection for a given learning algorithm is integrated into the training process. Through the wrapper technique, the features that are predicted by different learning algorithms with high predictive accuracy shall be chosen. The filter technique aims to classify a subset consisting of several features that are selected from the original set. Those features are selected

based on the evaluation criteria. This work aimed at employing wrapper and filter techniques because it takes the prediction capability into consideration. This shall lead to having results that are better than the results of other works. The work aimed at reducing the number of the features for increasing the detection rate, and the performance of the NIDS. Although scholars have proposed several NIDS models, the present study aimed to propose a model that is based on four well-known bio-inspired metaheuristic algorithms: genetic algorithm (GA) [11–13], particle swarm optimization (PSO) [14–16], grey wolf optimizer (GWO) [17–19] and firefly optimization algorithm (FFA) [20,21]. The latter model is tested through using a support vector machine (SVM) [22–24], J48 (C4.5) [24–26] and ML classifier.

The researcher of the present study proposed a feature selection model for the NIDS. This model is based on the PSO, GWO, FFA and GA algorithms. It aims to enhance the performance of NIDS through reducing the number of the selected features. It was evaluated through using SVM and the J48 ML classifier.

The present study is significant because it aimed to:

1. Identify the optimal feature set that is in the UNSW-NB15 dataset. The present study aimed to do that based on the PSO, GWO, FFA and GA algorithms;
2. Propose a filtering-based feature selection model for the NIDS. The present study aimed to do that based on the PSO, GWO, FFA and GA algorithms. It aimed to do that to reduce the number of the selected features;
3. Determine the best combination between the PSO, GWO, FFA and GA algorithms. The present study aimed to determine that to filter the selected features that improve the performance of the detection mechanism.

The structure of the present study is identified below:

Section 2 provides a literature review about the use of bio-inspired metaheuristic algorithms for building an efficient NIDS. In Section 3, brief information is presented about the proposed model. In Section 4, brief information is presented about the used dataset. Section 5 offers a description for the performance evaluation metrics. Section 6 presents the results and discussion. As for Section 7, it presents the conclusion.

2. Related Works

ML is commonly used for classifying anomalies in an IDS. ML is defined as a collection of computational methods that employ training data for enhancing performance, making specific future predictions and gaining knowledge from data. Figure 2 shows the procedures taken for creating an ML application.

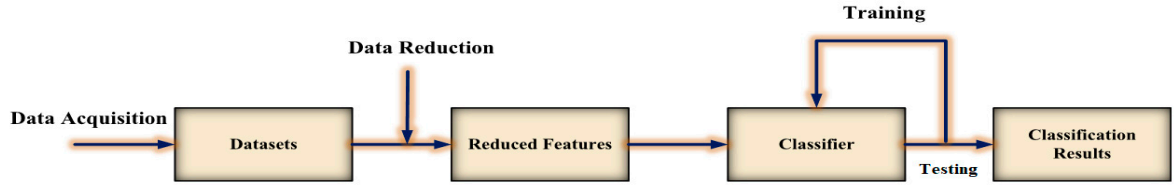

Figure 2. ML procedures [4].

Feature selection is a significant pre-processing stage in ML. It reduces the data dimensionality and increases the efficiency of the classification process. Scholars proposed several feature selection methods for IDSs. Those methods are proposed to classify important features based on several criteria. This section discusses briefly the state-of-the-art feature selection methods that are based on bio-inspired metaheuristic algorithms and ML classifiers. The latter methods aim to improve IDS performance.

Researchers in Reference [27] proposed a hybrid model of SVM along with GA for IDSs. This model can reduce the selected features from 41 features into 10 features. The selected features were categorized

into three categories—based on priority—through using GA. The features of the highest importance are considered first priority. The features of the lowest importance are considered third priority. The distribution of features was performed. For instance, four features are considered first priority, and four features are considered second priority. In addition, two features are considered third priority. The latter researchers employed the KDD'99 dataset in the experiment. It was found that the hybrid model can attain a positive detection of 0.973. It was found that the false alarm rate is 0.017.

Ahmad et al. [28] developed a feature selection model that is based on multilayer perception (MLP) for IDSs. This model is based on a combination of principal component analysis (PCA) and GA. The latter researchers conducted PCA to plan the features space to a principal feature space. They selected the features corresponding to the highest eigenvalues. The features that were selected by PCA may lack the adequate detection for the classifier. So, the researchers adopted GA to explore the principal feature space in order to find a subset with optimal sensitivity. The feature subsets selected through using PCA and GA were used to train the MLP classifier. The proposed method used the KDDCup'99 dataset in the evaluation. The number of the selected features was reduced from 41 features into 12 features only. The optimal features increased the detection accuracy. The latter accuracy became 99%.

Ghanem and Jantan [29] developed an artificial bee colony (ABC) method for the feature selection of IDSs. The latter method consists of two stages:

-Through stage 1, the subsets of features were generated through using the Pareto front non-dominated solutions;

-Through stage 2, a feed forward neural network (FFNN) and ABC (and PSO) were employed for assessing the feature subsets that were derived from the first stage.

Thus, the proposed method employs a new feature selection model. It is called (the multi-objective ABC method). It aims at reducing the number of network traffic features. The latter method adopts a new classification approach. The latter approach is named (the hybrid ABC-PSO approach). The latter method employs the optimized FFNN for categorizing the data derived from the first stage. Moreover, the latter researchers proposed a new fitness function for reducing the quantity of features. They did that to make sure that the false alarm rate is low.

Researchers in Reference [30] proposed a model to select features for IDSs. Those features were selected through using evolutionary algorithms: GA, PSO and differential evolution (DE). They conducted a comparison between these algorithms in terms of efficiency. They validated them through the use of the KDD Cup 99 data set, a neural network and an SVM. The optimum features that were selected by GA, PSO and DE were respectively as follows: 16, 15 and 13. They were selected from the 41 features that are in the dataset. They found that the training time of DE is 1.62 s. They found that DE is considered the best algorithm in terms of classification accuracy. To be specific, the classification accuracy of DE is 99.75%.

Researchers in Reference [31] proposed a new IDS model. The latter model is based on intelligent dynamic swarm through the use of a rough set. It is abbreviated as (IDS-RS) and simplified swarm optimization (SSO). It is considered as a new version of PSO that employs a new weighted local search (WLS) strategy. IDS-RS is conducted with a weighted sum fitness function to choose the most important features for having the features of the dataset reduced. They only collected six features out of 41 features that are located in the KDD99 dataset. At the final stage, the SSO classifier was used for identifying instances and achieving a classification accuracy of 93.3%.

The researcher in Reference [32] aimed to explore the performance level of the feature selection model that is in the NIDS. They aimed to explore that through using GA and PSO as algorithms for feature selection. GA and PSO played an effective role in having the number of the selected features reduced. The latter researchers found that GA can successfully reduce the number of the selected features from 41 features to 15 features. They found that PSO can have the number of the selected features successfully reduced from 41 features to 9 features. Through using k-nearest neighbor (k-NN) as a classifier, the GA-reduced dataset which consists of 37% of the original features shows

an improvement in accuracy from 99.28% to 99.70%. Through using k-nearest neighbor (k-NN) as a classifier, the GA-reduced dataset shows an execution time that is 4.8 times faster than the execution time of the original dataset. Through using the same classifier, PSO—which consists of 22% of the original features—shows the fastest execution time (7.2 times faster than the execution time of the original datasets). However, its accuracy is slightly reduced from 99.28% to 99.26%.

Researchers in Reference [18] employed the grey wolf optimization (GWO) method to search the feature space to find the optimal feature subset that improves the classification accuracy. The latter method used mutual information and filter-based principles. Second, the wrapper approach was adopted to raise the accuracy of the classifiers. Regarding the proposed approach, its accuracy was measured. The accuracy of the proposed approach was compared against the accuracy of several metaheuristic algorithms that employ the NSL-KDD dataset.

Researchers in Reference [21] used the firefly algorithm based on the filter and wrapper methods to select the features. They also proposed a procedure for raising the dimensionality. They used the wrapper ensemble method with the Bayesian network, C4.5 and mutual information (MI). Originally, the KDDCUP 99 dataset possessed 41 features. However, that approach reduced those features into 10 features. This reduced the computational cost of the classifier.

Al-Yaseen [33] proposed a wrapper feature selection method through employing the SVM and firefly algorithms. The proposed method improves the performance level of the intrusion detection system. It improves this through having the irrelevant features removed. It improves this through reducing the duration needed for the classification. It reduces this duration through having the dimensions of the data reduced. The latter researchers employed NSL-KDD along with employing the common measures of the intrusion detection systems. Such measures include: the overall accuracy, the rate of detection and the rate of false alarm. The proposed method achieved an overall accuracy of 78.89%. It was found that the proposed feature selection method is effective in enhancing the performance of the network intrusion detection system.

This work aimed to shed a light on the role of the MI of GA, PSO, GWO and FFA in finding the optimal set of features for NIDSs. It aimed to do that based on the UNSW-NB15 datasets. The ML of POS, GA, GWO and FFA algorithms was not considered in any of the works that address NIDSs. The section below offers information about the proposed model.

3. The Proposed Model

The proposed feature selection model aims at enhancing the performance of NIDSs. During recent years, numerous researchers employed data mining and ML techniques to solve problems and optimize system performance. This work has used the latter technique and reduced the number of features to raise the efficiency of NIDSs. Figure 3 presents the architecture of the model that has been proposed. The following subsection identifies the stages of the proposed model in details.

3.1. The Pre-Processing Stage

Through providing more appropriate data for the EvoloPy-FS optimization framework [34,35], the UNSW-NB15 dataset passed through several pre-processing steps. Those steps are identified below:

A The removal of the labels: Each feature in the original UNSW-NB15 dataset has a label. Removing those labels is important in order to adapt the dataset with the EvoloPy-FS environment;

B Removing Features: The original UNSW-NB15 dataset has 45 features. Two features of those features are class labels (attack cat and label). The attack cat cannot be considered as a feature. Thus, it is important to delete it. Deleting it is important because the main objective sought from this work is represented in reducing the features;

C Label encoding: Some labels in the dataset—e.g., protocol, state and service type—are given string values. Therefore, it is very significant to have those values encoded into numerical values;

D Data binarization: The numerical data in the dataset are in various ranges. During the training process, these data provide the classifier with a variety of challenges in order to compensate for such variations. Therefore, the values in each feature must be standardized. Thus, the least value in each one of the features should be 0. However, the maximum value should be 1. It makes the classifier more homogeneous. It preserves the difference between the values of each feature.

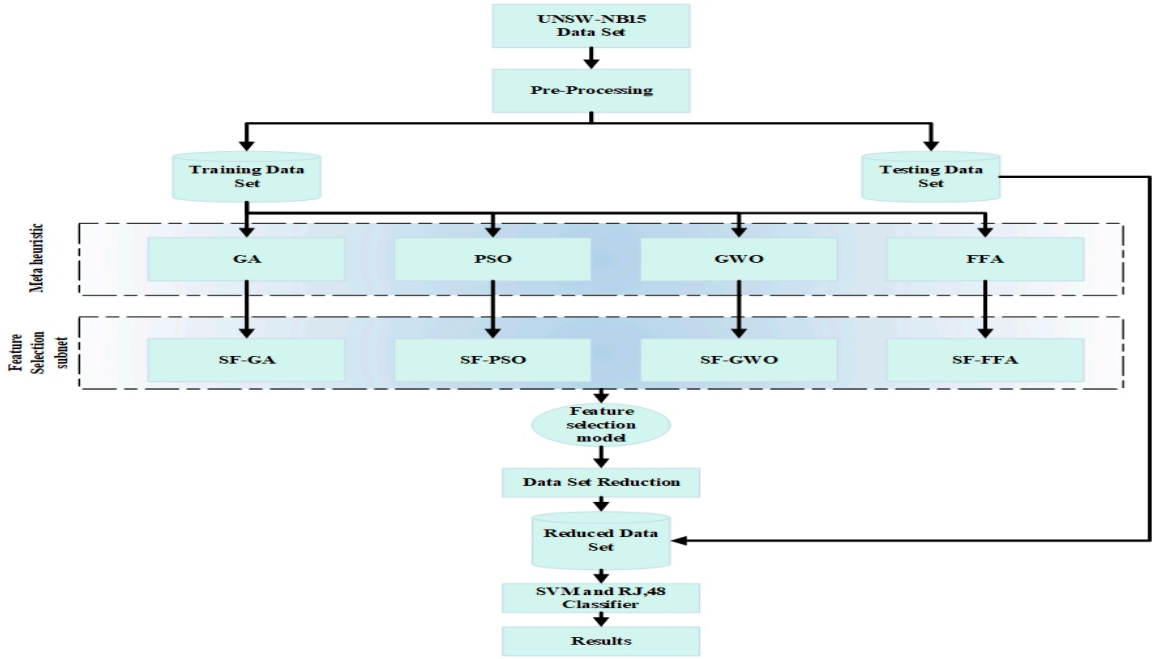

Figure 3. The architecture of the proposed model.

3.2. The Selection of Features Based on the Bio-Inspired Metaheuristic Algorithms

The selection of subset features is a difficult challenge. It cannot be managed efficiently when the dimensionality of the feature is high. Bio-inspired metaheuristic algorithms are suitable for addressing this challenge. They can offer high-quality solutions within an acceptable duration and through exerting reasonable effort. Through the proposed model, four subsets were extracted based on the GA, PSO, GWO and FFA algorithms.

3.2.1. GA Features Selection

GA [11–13] is an evolutionary search method that is employed for addressing the optimization problems based on a natural selection method. GA encodes a set of solutions for addressing the optimization problem. Those solutions are randomly generated to form a population. Then, GA evaluates this population in terms of a fitness function. The best solution is selected based on the problem being solved. It is assessed in terms of accuracy, root mean squared error (RMSE), F-measure or the area under curve (AUC). The fitter individuals were chosen for a set of reproduction operations, which are crossover and mutation. This operation gets repeated until it meets the termination criterion. This shall lead to forming a set of generations.

3.2.2. PSO Features Selection

Particle swarm optimization (PSO) was developed by Russell Eberhart and James Kennedy [36]. It was developed based on a simple concept derived from the movement of bird flocks and fish schools. It was developed after making several interpretations through using computer simulations.

PSO employs a variety of agents (particles) that make up a swarm. This swarm travels around in the search space in order to find the solution deemed the best. Regarding each particle in the search space, it alters its "flying" to match its flying experience and other particles' flying experience.

PSO is launched by randomly generated particles and their velocity, which indicate the search speed. Then, similar to the GA algorithm, the particles are evaluated in terms of fitness. Such evaluation is followed by two main tests. The first test compares the experience of a particle with itself, which is called personal best (pbest). The second test compares the fitness of a particle with the whole swarm experience. It is called global best (gbest). Performing these two tests leads to saving the best particle. After that, the termination criterion is met.

3.2.3. GWO Features Selection

GWO was proposed by Mirjalili et al. [17]. It was developed through performing hunting procedures. It was developed based on the leadership skills of grey wolves. The social hierarchy of wolves is shown in Figure 4. It describes four kinds of wolves: beta, alpha, omega and delta.

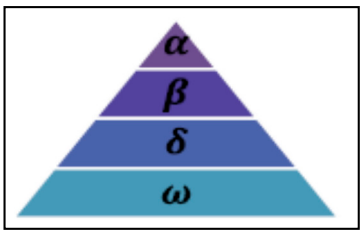

Figure 4. Wolves' hierarchy [17].

Alpha wolves are decision makers. They may not be the strongest in the pack, but they are certainly the best to manage the pack. This is because managing and organizing the pack are more significant than strength. Beta is a lower level wolf in the pack. It operates as an advisor to the alpha. It should be capable of taking the alpha's place in case of death or any other circumstances. Moreover, it reinforces the alpha's decisions among the members of the pack. It provides the alpha with the feedback of the members of the pack about the decision made by the alpha. Omega is deemed as the lowest level wolf among the pack. It acts as a scapegoat as the members of the pack submit to dominants. The existence of omega is very important. That is because omega preserves the dominant structure and satisfies the whole pack. Delta represents the rest of the pack which submits to beta and alpha. It includes: sentinels, scouts, elders, caretakers and hunters belonging to this level.

Based on this hierarchy, the group hunting process is performed through following three main steps. These steps are identified below:

(1) Tracking the prey and chasing and approaching it;
(2) Pursuing the prey and encircling and harassing it to stop its movement;
(3) Launching an attack against the prey being attacked. The algorithm mimics the whole described hierarchy and group hunting procedures. It mimics those procedures to solve complex engineering problems.

3.2.4. FFA Features Selection

The firefly optimization algorithm (FFA) for feature selection is a metaheuristic algorithm. It was proposed by Xin-She Yang [37]. It is based on tropical fireflies' communication behavior. It is also based on the idealized flashing pattern behavior. FFA employs the following idealized rules to construct the mathematical model of the algorithm.

(a) Regarding all the fireflies, they are unisex;
(b) The brightness of the fireflies is proportional to their attractiveness;
(c) The firefly's brightness is determined and influenced by the environment of the objective functions. In terms of the maximization problem, the brightness may be proportional to the value of the objective function.

The regular firefly algorithm includes two significant points. The first point is the formulation of the light intensity. The second point is the shift in attractiveness. One can always presume that the encoded objective feature landscape shall determine the brightness of the firefly. One should describe the light intensity difference and formulate the attractiveness adjustment.

3.3. Feature Selection Model Based on MI

The feature selection's set of bio-inspired metaheuristic algorithms are described as follows:

- Selected feature set based on PSO (S1);
- Selected feature set based on GWO (S2);
- Selected feature set based on FFA (S3);
- Selected feature set based on GA (S4).

One subset from those selected feature sets is generated based on MI using different rules as displayed in Table 1.

Table 1. The rules of the proposed model.

Rule Number	Rules	Output
R1	$S\{f: f \in (S1 \cap S2)\}$	S5
R2	$S\{f: f \in ((S1 \cap S3)\}$	S6
R3	$S\{f: f \in ((S1 \cap S4)\}$	S7
R4	$S\{f: f \in ((S2 \cap S3)\}$	S8
R5	$S\{f: f \in ((S2 \cap S4)\}$	S9
R6	$S\{f: f \in ((S3 \cap S4)\}$	S10
R7	$S\{f: f \in ((S1 \cap S2 \cap S3)\}$	S11
R8	$S\{f: f \in ((S1 \cap S2 \cap S4)\}$	S12
R9	$S\{f: f \in ((S1 \cap S3 \cap S4)\}$	S13
R10	$S\{f: f \in ((S2 \cap S3 \cap S4)\}$	S14
R11	$S\{f: f \in ((S1 \cap S2 \cap S3 \cap S4)\}$	S15
R12	$S\{f: f \in ((S11 \cap S12 \cap S13 \cap S14)\}$	S16
R13	$S\{f: f \in ((S5 \cap S6 \cap S7 \cap S8 \cap S9 \cap S10)\}$	S17

3.4. Machine Learning Classifiers

MLCs are used in ML for classifying data. Therefore, the output resulting in the feature set based on the proposed model rules is used as the input to the ML classifier. The function of the classifier is represented in classifying the incoming data as normal or abnormal data. In the present study, SVM and the J48 classifier are used.

3.4.1. SVM Classifier

SVM is a binary classifier. It is a common approach for making classifications between two classes. In SVM, a hyper plan is created to distinguish the positive sample class from the negative sample class based on the structural risk minimization principle. Alternatively, by choosing from different kernel functions, SVM can solve the problems of linear classification. SVM can get extended to nonlinear classification cases. It is a significant classification ML method because it employs the statistical learning theory [38]. Furthermore, due to its use for the structure risk minimization method, SVM has a strong generalization capability. Hence, SVM can be seen as a method that is better and more effective than another possible classifier. Through reviewing the relevant works that shed a light on IDSs [27,39], it has been proved that SVM is an effective classifier and shows a performance that is better than other classifiers.

3.4.2. J48 (C4.5 Decision Tree) Classifier

The J48 algorithm is the most popular tree classifier. It was developed by Quinlan [40]. It is an ID3 algorithm extension which uses a predictive ML model. The J48 algorithm uses the improved tree pruning technique to reduce the number of classification errors. In addition, the J48 algorithm adopts a dividing-and-conquer greedy approach for inducing recursively the decision trees that contain the features of the dataset for performing an additional classification. The J48 classifier algorithm is divided into datasets based on the attribute values of data to distinguish the probable prediction. The J48 classification algorithm will build its decision tree based on the theoretical attribute values of the present training data. Furthermore, in the J48 algorithm, each feature calculates the gain value separately. The estimation process proceeds until the process of prediction gets completed. A suitable feature is defined as the function that offers much information about the data instances. Several studies aimed at exploring the impact of using the J48 algorithm to improve the accuracy of IDSs [41].

4. Dataset Description

The dataset performs a major function in testing an IDS and measuring its performance. During the last couple of decades, numerous IDS datasets were introduced. Such datasets include: DARPA Dataset, KDDCup99, NSL-KDD and UNSW-NB15. A dataset typically consists of several attributes. These attributes are named class and feature. Most of the studies that shed a light on IDSs employed KDDCup99 and NSL-KDD. This study used the UNSW-NB15 dataset because the KDDCup99 and NSL-KDD datasets cannot meet the requirements of the current study. That is attributed to the rapid development of network security and the need for meeting operational requirements. While having inherent vulnerabilities in the dataset, they do not have typical traffic of the modern day nor do they have modified patterns of attack.

The UNSW-NB15 dataset was developed recently by Moustafa et al [42]. Figure 5 shows UNSW-NB15 testbed. The UNSW-NB15 dataset is a hybrid dataset that consists of an actual current normal network operation and synthetic modified attack. The researcher of the present study employed the UNSW-NB15 dataset in this research. The UNSW-NB15 dataset was created through using IXIA PerfectStorm, an attack generation tool. It contains nine families of modified attacks and real ones. These attacks are launched against different servers. The authors obtained tcpdump traces of the network traffic at the beginning of 2015 for a total period of 31 h. They created a dataset from these network logs, which consists of 49 features for each network flow.

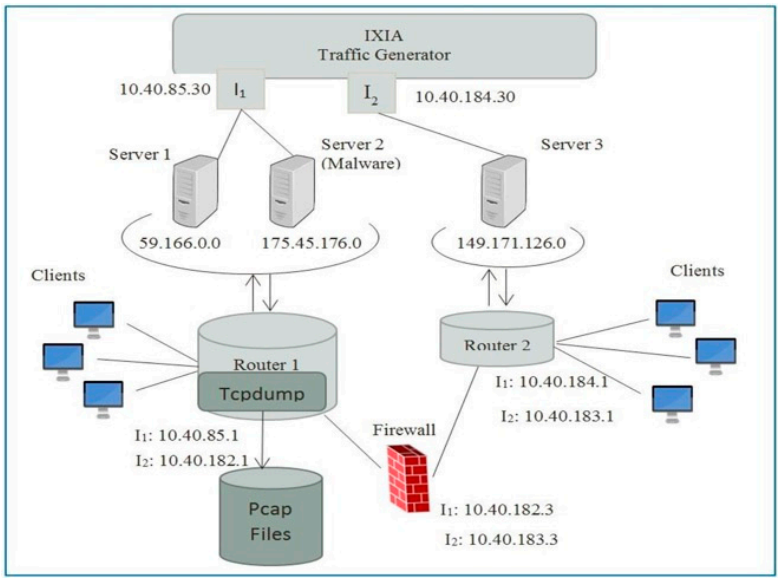

Figure 5. UNSW-NB15 testbed [42].

Support is received from Argus, Bro-IDS and custom utilities. Through such support, the features are extracted during the development process of the UNSW-NB15 dataset. They feed the pcap files into Bro-IDS and Argus. Regarding Argus, it is capable of handling raw traffic in the network. It operates on a client–server model, where the Argus server converts raw pcaps files into a format consistent with Argus. The Argus client will then read the functions and extract them from the Argus scripts. For every data instance, 49 connection features are available. Some of the features are statistical and other features are numerical. Other features suggest time stamp values. The UNSW-NB15 dataset got split into two datasets, testing datasets and training datasets. In the training set, there are 175,341 records. In the testing dataset, there are 82,332 records including all forms of attacks and usual traffic recordings. The testing datasets and training datasets have 45 features. Those features are listed in Table 2. There are features missing in the UNSW-NB15 training and testing datasets. These features are: ltime, sport, scrip, stime and dstip.

Table 2. Features Listed in UNSW-NB15 Dataset.

Feature No	Feature Name	Feature No	Feature Name	Feature No	Feature Name
1	id	16	dloss	31	response_body_len
2	dur	17	sinpkt	32	ct_srv_src
3	proto	18	dinpkt	33	ct_state_ttl
4	service	19	sjit	34	ct_dst_ltm
5	state	20	djit	35	ct_src_dport_ltm
6	spkts	21	swin	36	ct_dst_sport_ltm
7	dpkts	22	stcpb	37	ct_dst_src_ltm
8	sbytes	23	dtcpb	38	is_ftp_login
9	dbytes	24	dwin	39	ct_ftp_cmd
10	rate	25	tcprtt	40	ct_flw_http_mthd
11	sttl	26	synack	41	ct_src_ltm
12	dttl	27	ackdat	42	ct_srv_dst
13	sload	28	smean	43	is_sm_ips_ports
14	dload	29	dmean	44	attack_cat
15	sloss	30	trans_depth	45	label

5. Performance Evaluation Metrics

For assessing the efficiency level of the proposed model, the following metrics employ several features. These metrics are: true positive (TP), true negative (TN), false positive (FP) and false negative (FN) [43]. The confusion matrix—as displayed in Table 3—calculates true positive rate (TPR), true negative rate (TNR), false positive rate (FPR) and false negative rate (FNR). Based on these metrics, other factors may be derived. Such factors include: sensitivity, precision, accuracy and F-measure.

Table 3. Confusion matrix.

		Predicted	
		Normal	**Attack**
Actual	Normal	a (TP)	b (FN)
	Attack	c (FP)	d (TN)

TPR is measured for estimating the quantity of the normal data identified as being normal data. It is calculated as follows:

$$TPR = \frac{a}{a+b} \tag{2}$$

TNR is measured for estimating the quantity of the attack data identified as being attack data. It is calculated as follows:

$$TNR = \frac{d}{d+c} \tag{3}$$

FPR is measured for estimating the quantity of the attack data that is identified as being normal data. It is calculated as follows:

$$FPR = \frac{c}{c+d}. \tag{4}$$

FNR is measured for estimating the quantity of the normal data that is identified as being attack data. It is calculated as follows:

$$FNR = \frac{b}{a+b} \tag{5}$$

Accuracy is represented in a percentage. It refers to the degree to which the instances are predicted correctly. It is calculated as follows:

$$Accuracy = \frac{TPR + TNR}{TPR + TNR + FPR + FNR}. \tag{6}$$

Precision is represented by the ratio of the number of decisions that are considered correct. It is represented in the TP divided by the sum of FP and TP. It is calculated as follows:

$$\text{Precision} = \frac{TPR}{TPR + FPR}. \tag{7}$$

Sensitivity is represented in the number of TP evaluations that is divided by the number of all of the positive evaluations. It is calculated as follows:

$$\text{Sensitivity } = \frac{TPR}{TPR + FNR}. \tag{8}$$

The F-measure serves as a measure for testing the level of accuracy. It refers to the balance existing between sensitivity on the one hand and precision on the other. It is calculated as follows:

$$\text{F} - \text{Measure } = \frac{2 * \text{Precision} * \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}} \tag{9}$$

6. Results and Discussion

6.1. Selected Feature Experiments Results

All experiments were performed on a 3.40 GHZ, i7 CPU, 6.0 GB RAM and Windows 7 operating system. The Anaconda Python Open Source [44] was used to make the experiments. Table 4 presents the selected features that are considered important based on the proposed model rules for detecting the attacks.

Table 4. Important selected features.

Rule	Select Features	Features Number
PSO	**f2,f4,f5,f7,f11,f12,f16,f17,f18,f19,f20,f22,f23,f24 f25,f26,f28,f30,f31,f33,f34,f39,f40,f41,f43**	**25**
GWO	**f1,f4,f5,f6,f9,f13,f16,f17,f22,f23,f26,f28,f29,f35, f36, f37,f38,f40,f41,f43**	**20**
FFA	**f1, f2, f3,f6,f8,f9,f10,f11,f12,f13,f16,f19,f26,f28, f31,32, f34,f35,f37,f41,f43**	**21**
GA	**f1,f2,f3,f4,f6,f7,f8,f9,f11,f16,f21,f24,f25,f27,f28 f32,f34,f35,f37,f39,f41,f42,f43**	**23**
(R1) PSO∩ GWO	**f4,f5,f16,f17,f22,f23,f26,f28,f35,f40,f41,f43**	12
(R2) PSO∩FFA	**f2,f11,f12,f16,f19,f26,f28,f31,f35,f41,43**	11
(R3) PSO∩GA	**f2,f4,f7,f11,f16,f24,f25,f28,f35,f39,41,f43**	12
(R4) GWO∩FFA	**f1,f6,f9,f13,f16,f26,f28,f35,f37,f41,f43**	11
(R5) GWO∩GA	**f1,f4,f6,f9,f16,f28,f35,f37,f41,f43**	10
(R6) FFA∩GA	**f1,f2,f3,f6,f8,f9,f11,f16,f28,f32,f34,f35,f37,f41,f43**	15
(R7) PSO∩GWO∩FFA	**f16,f26,f28,f35,f42,f43**	6
(R8) PSO∩GWO∩GA	**f4,f16,f28,f35,f41,f43**	6
(R9) PSO∩FFA∩GA	**f2,f11,f16,f28,f35,f41,f43**	7
(R10) GWO∩FFA∩GA	**f1,f6,f9,f16,f28,f38,f37,f41,f43**	9
(R11) PSO∩GWO∩FFA∩GA	**f16,f28,f35,f41,f43**	5
(R12) (PSO∩GWO∩FFA)∪ (PSO∩GWO∩GA) ∪ (PSO∩ FFA ∩GA)∪ (GWO∩FFA ∩GA)	**f1,f2,f4,f6,f9,f11,f16,f26,f28,f35,f37,f41,f43**	13
(R13) (PSO∩GWO)∪(PSO∩FFA)∪ (PSO∩GA)∪(GWO∩FFA)∪ (GWO∩GA)∪(FFA∩GA)	**f1,f2,f3,f4,f5,f6,f7,f8,f9,f11,f12,f13,f16,f17 f19,f22,f23,f24,f25,f26,f28,f31,f32 f34,f35,f37,f39,f40,41,f43**	30

6.2. Experimental Evaluation Results

The proposed model is evaluated based on the J48 and SVM ML classifiers. The outcomes of the experiment that is based on the J48 classifier are presented in Table 5. The results of the experiment that is based on SVM are presented in Table 6. The rates of the classification accuracy of the proposed approach are within the range of 79.175%–90.484% based on the J48 classifier. The rates of the classification accuracy of the proposed approach are within the range of 79.077%–90.119% based on the SVM classifiers.

The accuracies of most of the reduction rules of the proposed models in this paper are higher than the accuracy of all the features. All the algorithms and rules were evaluated in terms of TPR, FNR, TNR, FPR, accuracy, sensitivity, precision and F1-measure. It was found that the rules of the proposed model vary in terms of effectiveness.

Table 5. The results reached based on J48.

Rule	TPR	FNR	FPR	TNR	Accuracy	Precision	Sensitivity	F-Measure
All	63.99%	36.01%	4.57%	95.42%	81.29%	91.94%	63.98%	75.46%
PSO	80.844%	19.156%	2.817%	97.183%	89.013%	96.107%	80.844%	87.817%
GWO	93.797%	6.203%	20.952%	79.048%	85.676%	78.513%	93.797%	85.477%
FFA	96.586%	3.414%	22.592%	77.408%	86.037%	77.764%	96.586%	86.159%
GA	96.700%	3.300%	21.164%	78.836%	86.874%	78.892%	96.700%	86.893%
R1	81.097%	18.903%	22.331%	77.669%	79.210%	74.774%	81.097%	77.807%
R2	93.854%	6.146%	24.307%	75.693%	83.854%	75.912%	93.854%	83.935%
R3	94.124%	5.876%	24.307%	75.693%	83.976%	75.965%	94.124%	84.075%
R4	94.314%	5.686%	24.307%	75.693%	84.061%	76.001%	94.314%	84.173%
R5	94.314%	5.686%	23.204%	76.796%	84.668%	76.838%	94.314%	84.684%
R6	94.314%	5.686%	22.984%	77.016%	84.790%	77.008%	94.314%	84.786%
R7	86.481%	13.519%	24.537%	75.463%	80.415%	74.205%	86.481%	79.874%
R8	86.349%	13.651%	26.681%	73.319%	79.175%	72.539%	86.349%	78.844%
R9	86.349%	13.651%	26.681%	73.319%	79.175%	72.539%	86.349%	78.844%
R10	96.549%	3.451%	25.410%	74.590%	84.458%	75.617%	96.549%	84.810%
R11	89.051%	10.949%	26.681%	73.319%	80.389%	73.148%	89.051%	80.320%
R12	97.127%	2.873%	16.587%	83.413%	89.576%	82.697%	97.127%	89.333%
R13	97.141%	2.859%	14.950%	85.050%	90.484%	84.136%	97.141%	90.172%

Table 6. The results reached based on SVM.

Rules	TPR	FNR	FPR	TNR	Accuracy	Precision	Sensitivity	F-Measure
All	63.965%	36.035%	4.809%	95.191%	81.158%	91.566%	63.965%	75.316%
PSO	79.562%	20.438%	2.596%	97.404%	89.152%	96.345%	79.562%	87.153%
GWO	93.570%	6.430%	22.931%	77.069%	84.485%	76.908%	93.570%	84.425%
FFA	95.235%	4.765%	22.592%	77.408%	85.429%	77.519%	95.235%	85.469%
GA	96.970%	3.030%	22.270%	77.730%	86.387%	78.079%	96.970%	86.505%
R1	80.827%	19.173%	23.265%	76.735%	78.576%	74.774%	80.827%	77.248%
R2	93.884%	6.116%	24.994%	75.006%	83.388%	74.998%	93.884%	83.385%
R3	94.154%	5.846%	24.994%	75.006%	83.508%	75.052%	94.154%	83.525%
R4	94.154%	5.846%	24.562%	75.438%	83.748%	75.377%	94.154%	83.726%
R5	94.154%	5.846%	24.346%	75.654%	83.868%	75.540%	94.154%	83.826%
R6	94.154%	5.846%	24.130%	75.870%	83.988%	75.705%	94.154%	83.927%
R7	86.751%	13.249%	24.978%	75.022%	80.293%	73.923%	86.751%	79.825%
R8	86.403%	13.597%	26.902%	73.098%	79.077%	72.387%	86.403%	78.776%
R9	86.403%	13.597%	26.902%	73.098%	79.077%	72.387%	86.403%	78.776%
R10	96.278%	3.722%	25.631%	74.369%	84.215%	75.405%	96.278%	84.573%
R11	88.781%	11.219%	26.902%	73.098%	80.146%	72.926%	88.781%	80.077%
R12	96.586%	3.414%	16.587%	83.413%	89.333%	82.617%	96.586%	89.058%
R13	96.870%	3.130%	15.391%	84.609%	90.119%	83.706%	96.870%	89.808%

Based on the TPR, the data are normal and identified as normal. Figure 6 presents the TPR for all the features and the rules based on J48 and SVM.

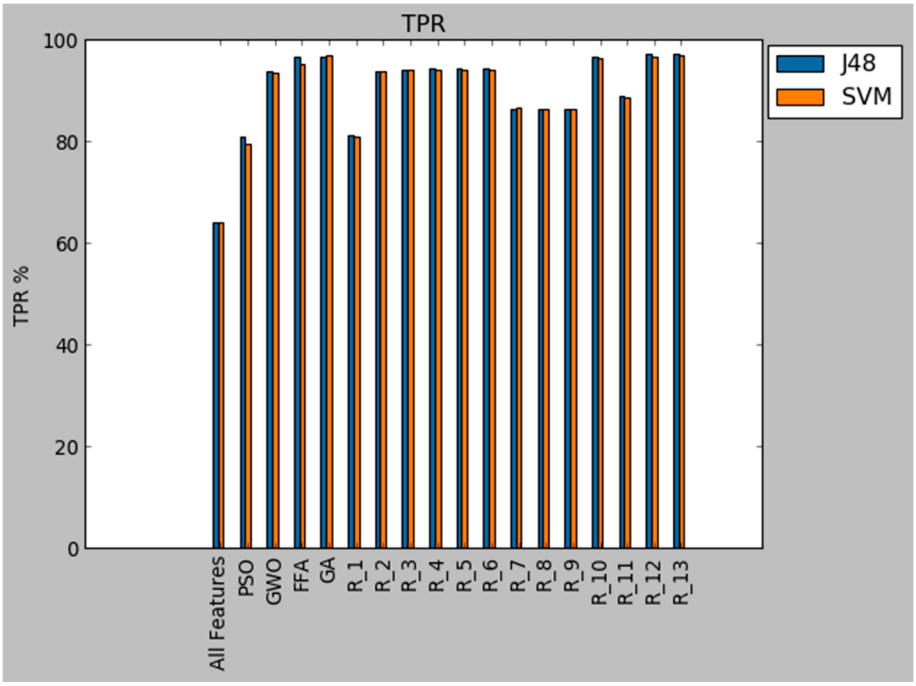

Figure 6. True Positive Rate (TPR).

The TPRs of the features based on the J48 and SVM classifiers are 63.99% and 63.96%, respectively. The performance of J48 is a little better than the performance of SVM. GA has the highest TPR based on the SVM classifier. R13 shows the highest TPR based on the J48 classifier.

The FNR indicates that the data are normal and identified as attack. Figure 7 shows the FNR results for SVM and J48.

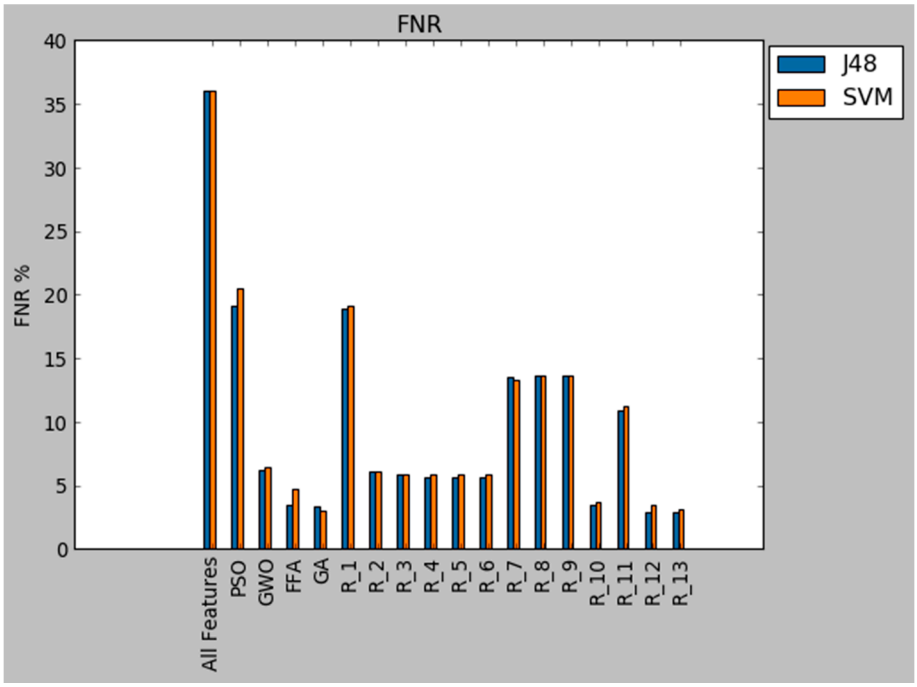

Figure 7. False Negative Rate (FNR).

The highest FNR was obtained from all features based on J48 and SVM. GA has the lowest FNR based on the SVM classifier. R13 has the lowest FNR based on the J48 classifier.

The FPR indicates that the data are attack and identified as normal. Figure 8 shows the FPR results based on SVM and J48.

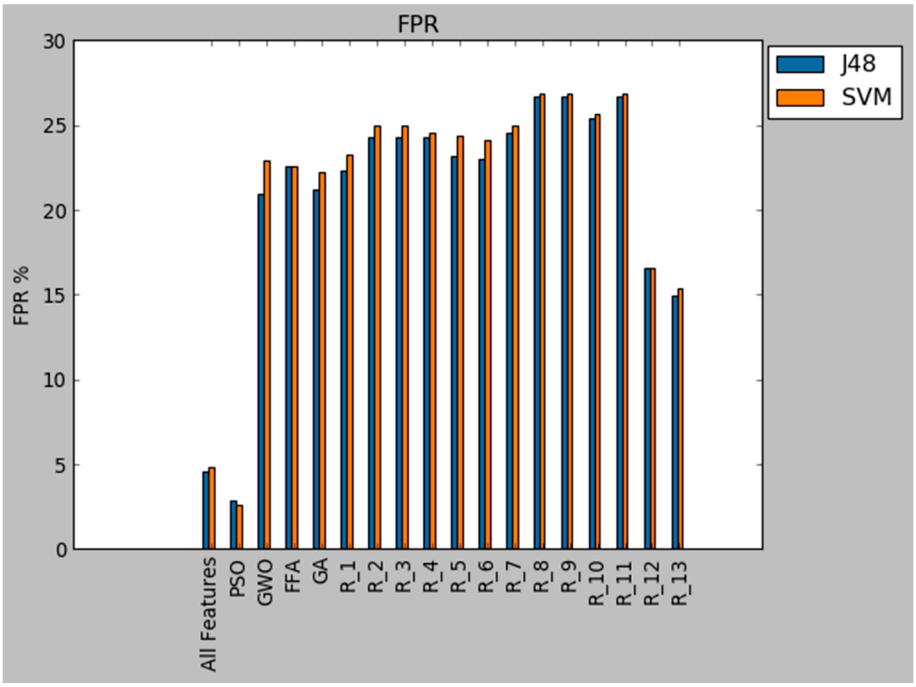

Figure 8. False Positive Rate (FPR).

The highest FPR was obtained from R11 and R9 and R8 for J48 and SVM, respectively. PSO shows the lowest FPR based on the J48 and SVM classifiers.

The TN indicates that the data are attack data and identified as attack. Figure 9 shows the TNR results of the proposed model based on SVM and J48.

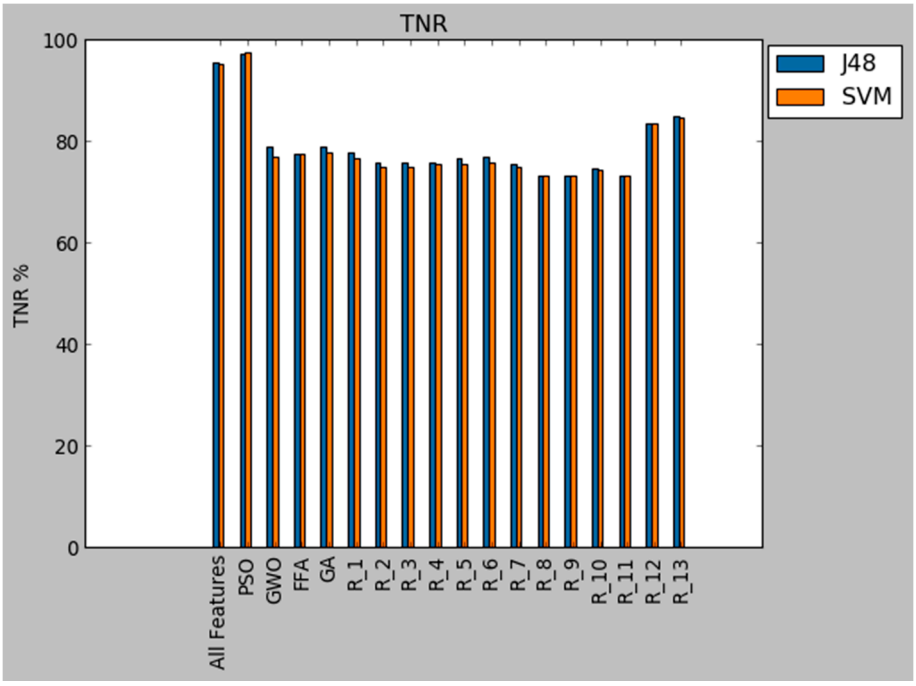

Figure 9. True Negative Rate (TNR).

The highest TNR was obtained from PSO for J48 and SVM. R11 and R9 and R8 obtained the lowest rates for J48 and SVM, respectively.

The accuracy reflects how accurate the process of classifying the normal and anomalous behaviors is. It is calculated as the percentage of the correctly categorized data in all dataset ranges. Figure 10 shows the accuracy of the proposed model based on SVM and J48.

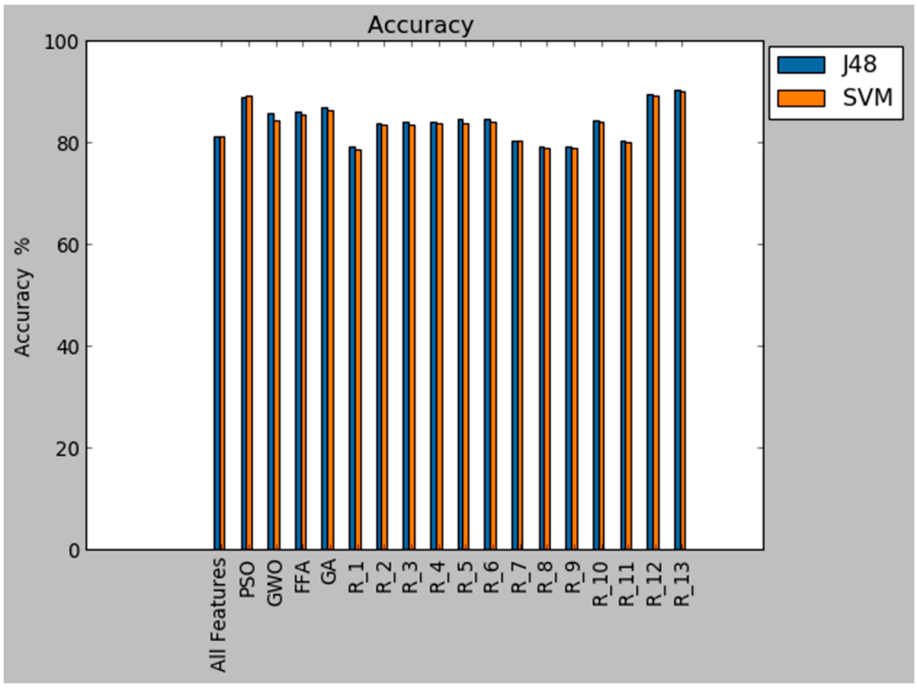

Figure 10. Accuracy.

The results in Figure 10 show that R13 and R12 have the highest accuracy. R13 shows an accuracy of 90.48% based on J48. It shows an accuracy of 90.12% based on SVM. R13 reduced the number of features into 30 features. R12 shows an accuracy of 89.58% and 89.33% based on J48 and SVM, respectively. R12 reduced the number of features into 13 features.

Precision refers to the ratio of the truly positive to all the positive results. Figure 11 shows the precision of the proposed model based on SVM and J48. It was found that PSO shows the highest precision.

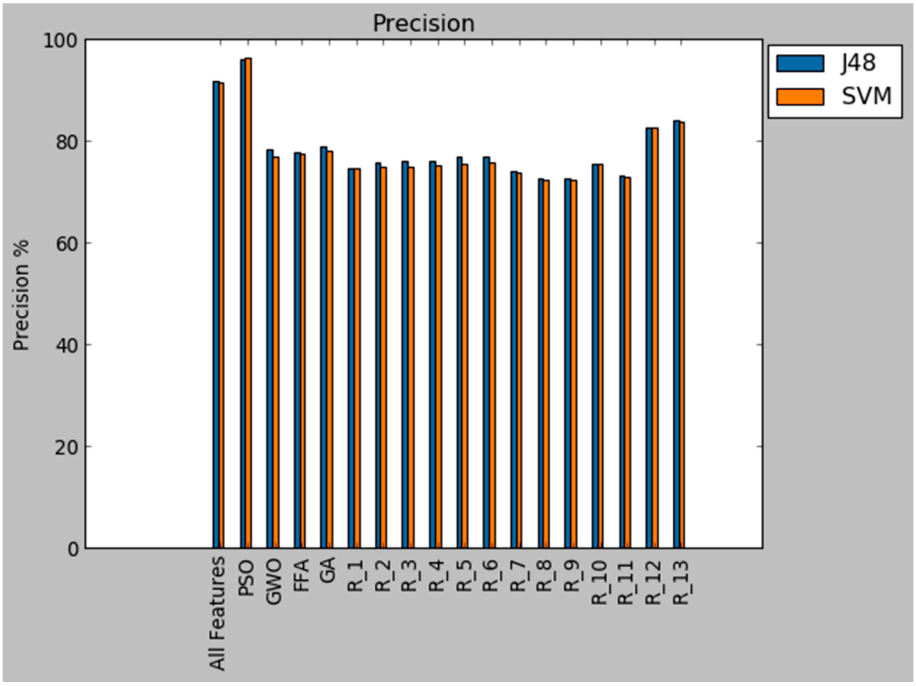

Figure 11. The precision rate.

Figure 12 shows the sensitivity rate of the proposed model based on J48 and SVM. Sensitivity reflects the ability of an IDS to identify a relation as being an attack.

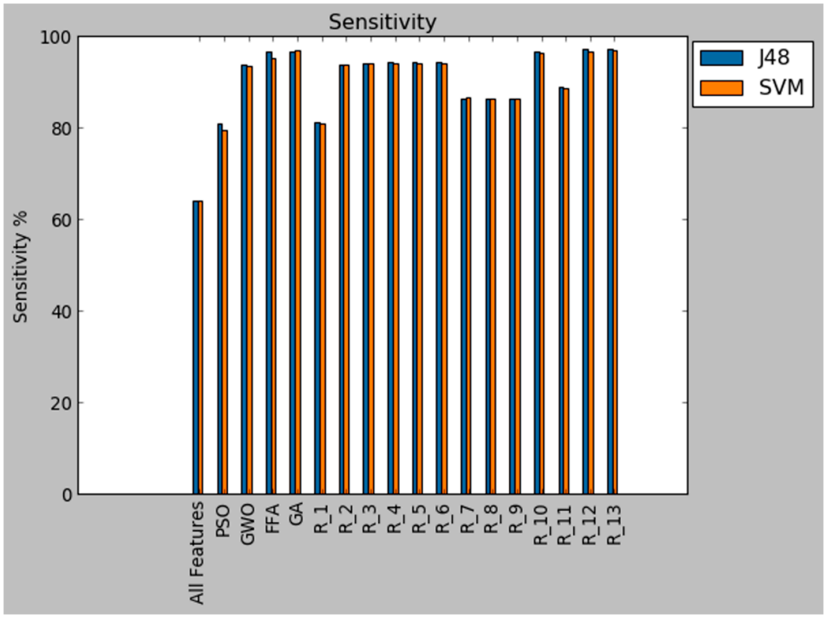

Figure 12. The sensitivity rate.

The results in Figure 12 show that R13 and R12 show the best sensitivity rates. The other features show the worst sensitivity rates based on both classifiers. This means that R13 and R12 can detect anomalies effectively at higher rates.

F-measure considers precision and sensitivity. F-measure acts as a balance between precision and sensitivity. It serves as the harmonious measure of sensitivity and precision. Figure 13 shows the F-measure values for the proposed model based on J48 and SVM.

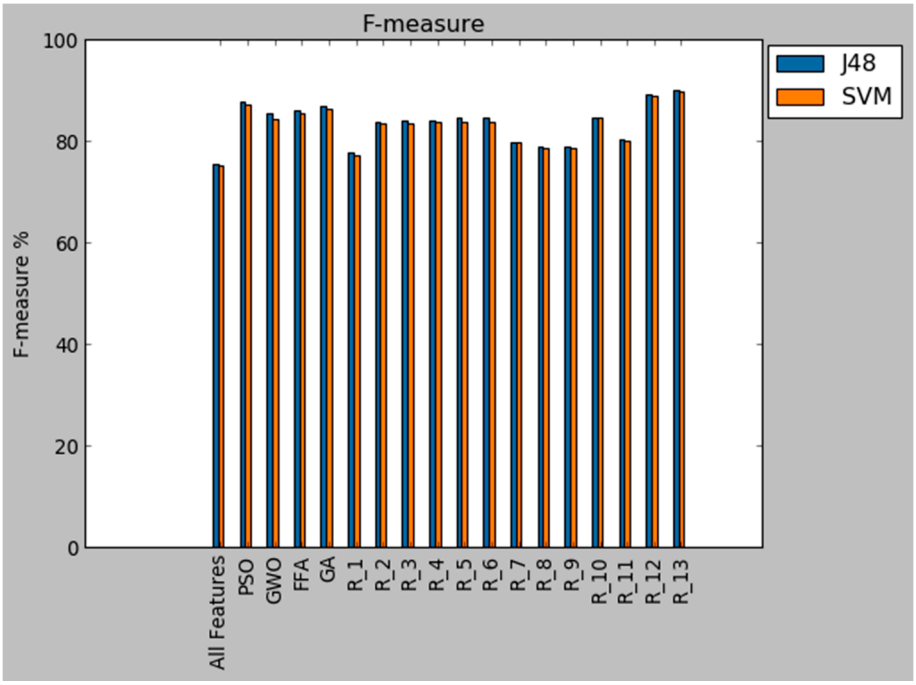

Figure 13. The F-measure rate.

Based on the results above, R13 and R12 show the best F-measure rates, whereas, the other features have the worst F-measure rate. It can be observed that all the evaluation metrics can evaluate the quality of IDSs. As for accuracy and the F-measure rate, they can be used to evaluate on the overall efficiency of the NIDS.

Based on all experiments, R13 and R12 provided the best results in terms of the F-measure, accuracy and sensitivity based on the J48 and SVM classifiers. GA shows good results in terms of FNR and TPR, whereas H11, H9 and H8 show good results in terms of FPR. PSO shows good results in terms of precision and TNR.

7. Conclusions

Improving an intrusion detection system is something which is challenging. The detection rate of an NIDS is affected by the number of features. The key task of data mining and ML techniques aim at improving the detection accuracy and reducing the positive false rate for an NIDS. The latest models failed to identify the network intrusion through using all the UNSW-NB15 dataset features. The researcher of the present study aimed to propose an NIDS model that contains 17 rules for feature selection. The latter model is based on the UNSW-NB15 dataset. The proposed feature selection model is based on the PSO, GWO, FFA and GA bio-inspired algorithms and MI. In the case of bio-inspired algorithms, PSO reduces the number of the selected features to 25 features; GWO reduces the number of the selected features to 20 features; FFA reduces the number of the selected features to 21 features; and GA reduces the number of the selected features to 23 features. In the case of the MI of PSO, GWO, FFA and GA, R1 reduces the number of the selected features to 12 features; R2 reduces the number of the selected features to 11 features; R3 reduces the number of the selected features to 12 acceptable

features; R4 reduces the number of the selected features to 11 features; R5 reduces the number of the selected features to 10 features; R6 reduces the number of the selected features to 15 features; R7 reduces the number of the selected features to 6 features; R8 reduces the number of the selected features to 6 features; R9 reduces the number of the selected features to 7 features; R10 reduces the number of the selected features to 9 features; R11 reduces the number of the selected features to 5 features; R12 reduces the number of the selected features to 13 features; and R13 reduces the number of the selected features to 30 features.

R13 and R12 show the best results in terms of F-measure, accuracy and sensitivity based on the J48 and SVM ML classifiers. The researcher recommends conducting other studies for assessing the effectiveness of the proposed model using deep learning architectures, such as: the recurrent neural network (RNN) and convolutionary neural network (CNN).

Funding: This research received no external funding.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Vinchurkar, D.P.; Reshamwala, A. A Review of Intrusion Detection System Using Neural Network and Machine Learning. *J. Eng. Sci. Innov. Technol.* **2012**, *1*, 54–63.
2. Othman, S.M.; Alsohybe, N.T.; Ba-Alwi, F.M.; Zahary, A.T. Survey on Intrusion Detection System Types. *Int. J. Cyber Secur. Digit. Forensics* **2018**, *7*, 444–463.
3. Zarpelão, B.B.; Miani, R.S.; Kawakani, C.T.; de Alvarenga, S.C. A survey of intrusion detection in Internet of Things. *J. Netw. Comput. Appl.* **2017**, *84*, 25–37. [CrossRef]
4. Kwon, D.; Kim, H.; Kim, J.; Suh, S.C.; Kim, I.; Kim, K.J. A survey of deep learning-based network anomaly detection. *Clust. Comput.* **2017**, *22*, 1–13. [CrossRef]
5. Win, T.Z.; Kham, N.S.M. Information Gain Measured Feature Selection to Reduce High Dimensional Data. In Proceedings of the 17th International Conference on Computer Applications (ICCA 2019), Novotel hotel, Yangon, Myanmar, 27 February–1 March 2019; pp. 68–73.
6. Liu, H.; Motoda, H. *Feature Selection for Knowledge Discovery and Data Mining*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012; Volume 454.
7. Al-Tashi, Q.; Rais, H.M.; Abdulkadir, S.J.; Mirjalili, S.; Alhussian, H. A Review of Grey Wolf Optimizer-Based Feature Selection Methods for Classification. In *Evolutionary Machine Learning Techniques*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 273–286.
8. Emary, E.; Zawbaa, H.M.; Hassanien, A.E. Binary grey wolf optimization approaches for feature selection. *Neurocomputing* **2016**, *172*, 371–381. [CrossRef]
9. Al-Tashi, Q.; Kadir, S.J.A.; Rais, H.M.; Mirjalili, S.; Alhussian, H. Binary optimization using hybrid grey wolf optimization for feature selection. *IEEE Access* **2019**, *7*, 39496–39508. [CrossRef]
10. Sahoo, A.; Chandra, S. Multi-objective grey wolf optimizer for improved cervix lesion classification. *Appl. Soft Comput.* **2017**, *52*, 64–80. [CrossRef]
11. Mitchell, M. *An Introduction to Genetic Algorithms*; MIT Press: Cambridge, MA, USA, 1998.
12. Gharaee, H.; Hosseinvand, H. A new feature selection IDS based on genetic algorithm and SVM. In Proceedings of the 2016 8th International Symposium on Telecommunications (IST), Tehran, Iran, 27–28 September 2016; pp. 139–144.
13. Al Balas, F.; Almomani, O.; Jazoh, R.M.A.; Khamayseh, Y.M.; Saaidah, A. An Enhanced End to End Route Discovery in AODV using Multi-Objectives Genetic Algorithm. In Proceedings of the 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 9–11 April 2019; pp. 209–214.
14. Marini, F.; Walczak, B. Particle swarm optimization (PSO). A tutorial. *Chemom. Intell. Lab. Syst.* **2015**, *149*, 153–165. [CrossRef]
15. Srinoy, S. Intrusion detection model based on particle swarm optimization and support vector machine. In Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Security and Defense Applications, Honolulu, HI, USA, 1–5 April 2007; pp. 186–192.

16. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
17. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
18. Devi, E.M.; Suganthe, R.C. Feature selection in intrusion detection grey wolf optimizer. *Asian J. Res. Soc. Sci. Humanit.* **2017**, *7*, 671–682. [CrossRef]
19. Alzubi, Q.M.; Anbar, M.; Alqattan, Z.N.M.; Al-Betar, M.A.; Abdullah, R. Intrusion detection system based on a modified binary grey wolf optimisation. *Neural Comput. Appl.* **2019**, *32*, 6125–6137. [CrossRef]
20. Yang, X.-S.; He, X. Firefly algorithm: Recent advances and applications. *arXiv* **2013**, arXiv:1308.3898. [CrossRef]
21. Selvakumar, B.; Muneeswaran, K. Firefly algorithm based feature selection for network intrusion detection. *Comput. Secur.* **2019**, *81*, 148–155.
22. Hasan, M.A.M.; Nasser, M.; Pal, B.; Ahmad, S. Support vector machine and random forest modeling for intrusion detection system (IDS). *J. Intell. Learn. Syst. Appl.* **2014**, *6*, 42869. [CrossRef]
23. Mohammad, A.H.; Alwada'n, T.; Al-Momani, O. Arabic text categorization using support vector machine, Naïve Bayes and neural network. *Gstf J. Comput.* **2016**, *5*, 108. [CrossRef]
24. Madi, M.; Jarghon, F.; Fazea, Y.; Almomani, O.; Saaidah, A. Comparative analysis of classification techniques for network fault management. *Turk. J. Elec. Eng. Comp. Sci.* **2020**, *28*, 1442–1457. [CrossRef]
25. Sahu, S.; Mehtre, B.M. Network intrusion detection system using J48 Decision Tree. In Proceedings of the 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Kochi, India, 10–13 Augest 2015; pp. 2023–2026.
26. Mohammad, A.H.; Al-Momani, O.; Alwada'n, T. Arabic text categorization using k-nearest neighbour, Decision Trees (C4. 5) and Rocchio classifier: A comparative study. *Int. J. Curr. Eng. Technol.* **2016**, *6*, 477–482.
27. Aslahi-Shahri, B.M.; Rahmani, R.; Chizari, M.; Maralani, A.; Eslami, M.; Golkar, M.J.; Ebrahimi, A. A hybrid method consisting of GA and SVM for intrusion detection system. *Neural Comput. Appl.* **2016**, *27*, 1669–1676. [CrossRef]
28. Ahmad, I.; Abdullah, A.; Alghamdi, A.; Alnfajan, K.; Hussain, M. Intrusion detection using feature subset selection based on MLP. *Sci. Res. Essays* **2011**, *6*, 6804–6810.
29. Ghanem, W.; Jantan, A. Novel multi-objective artificial bee colony optimization for wrapper based feature selection in intrusion detection. *Int. J. Adv. Soft Comput. Appl.* **2016**, *8*, 70–81.
30. Zaman, S.; El-Abed, M.; Karray, F. Features selection approaches for intrusion detection systems based on evolution algorithms. In Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication, Kota Kinabalu, Malaysia, 17–19 January 2013; pp. 1–5.
31. Chung, Y.Y.; Wahid, N. A hybrid network intrusion detection system using simplified swarm optimization (SSO). *Appl. Soft Comput.* **2012**, *12*, 3014–3022. [CrossRef]
32. Syarif, I. Feature selection of network intrusion data using genetic algorithm and particle swarm optimization. *EMITTER Int. J. Eng. Technol.* **2016**, *4*, 277–290. [CrossRef]
33. Al-Yaseen, W.L. Improving Intrusion Detection System by Developing Feature Selection Model Based on Firefly Algorithm and Support Vector Machine. *IAENG Int. J. Comput. Sci.* **2019**, *46*, 534–540.
34. Khurma, R.A.; Aljarah, I.; Sharieh, A.; Mirjalili, S. EvoloPy-FS: An Open-Source Nature-Inspired Optimization Framework in Python for Feature Selection. In *Evolutionary Machine Learning Techniques*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 131–173.
35. Faris, H.; Aljarah, I.; Mirjalili, S.; Castillo, P.A.; Guervós, J.J.M. EvoloPy: An Open-source Nature-inspired Optimization Framework in Python. In Proceedings of the 8th International Joint Conference on Computational Intelligence, Porto, Portugal, 9–11 November 2016; pp. 171–177.
36. Kennedy, J.; Eberhart, R. PSO optimization. In Proceedings of the Proc. IEEE Int. Conf. Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume 4, pp. 1941–1948.
37. Yang, X.-S. Firefly algorithm. *Nat. Inspired Metaheuristic Algorithms* **2008**, *20*, 79–90.
38. Vapnik, V.N. An overview of statistical learning theory. *IEEE Trans. Neural Netw.* **1999**, *10*, 988–999. [CrossRef]
39. Nagar, P.; Menaria, H.K.; Tiwari, M. Novel Approach of Intrusion Detection Classification Deeplearning Using SVM. In *First International Conference on Sustainable Technologies for Computational Intelligence, 2020, Advances in Intelligent Systems and Computing*; Springer: Singapore, 2020; Volume 1045, pp. 365–381. Available online: https://link.springer.com/chapter/10.1007%2F978-981-15-0029-9_29#citeas (accessed on 1 March 2020).
40. Quinlan, J.R. *C4. 5: Programs for Machine Learning*; Elsevier: Amsterdam, The Netherlands, 2014.

41. Aljawarneh, S.; Yassein, M.B.; Aljundi, M. An enhanced J48 classification algorithm for the anomaly intrusion detection systems. *Clust. Comput.* **2019**, *22*, 10549–10565. [CrossRef]
42. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the 2015 military communications and information systems conference (MilCIS), Canberra, Australia, 10–12 November 2015; pp. 1–6.
43. Smadi, S.; Aslam, N.; Zhang, L. Detection of online phishing email using dynamic evolving neural network based on reinforcement learning. *Decis. Support Syst.* **2018**, *107*, 88–102. [CrossRef]
44. Duchesnay, E.; Löfstedt, T. *Statistics and Machine Learning in Python. Release 0.1*; Springer: Berlin/Heidelberg, Germany, 2018.