Open Access
2021, Vol. 18 No.2 (Suppl. June)

Baghdad Science Journal

P-ISSN: 2078-8665
E-ISSN: 2411-7986

DOI: http://dx.doi.org/10.21123/bsj.2021.18.2(Suppl.).0884

Performance Evaluation of Intrusion Detection System using Selected Features and Machine Learning Classifiers

Raja Azlina Raja Mahmood[1] *Amir Hossien Abdi*[1] *Masnida Hussin*[1]

Universiti Putra Malaysia, Malaysia.
Corresponding author: raja_azlina@upm.edu.my, avinar1368@gmail.com, masnida@upm.edu.my
ORCID ID: https://orcid.org/0000-0002-1169-4226 , https://orcid.org/0000-0003-1063-8502 , https://orcid.org/0000-0001-7727-1739

Received 28/3/2021, Accepted 11/4/2021, Published 20/6/2021

Abstract:

Some of the main challenges in developing an effective network-based intrusion detection system (IDS) include analyzing large network traffic volumes and realizing the decision boundaries between normal and abnormal behaviors. Deploying feature selection together with efficient classifiers in the detection system can overcome these problems. Feature selection finds the most relevant features, thus reduces the dimensionality and complexity to analyze the network traffic. Moreover, using the most relevant features to build the predictive model, reduces the complexity of the developed model, thus reducing the building classifier model time and consequently improves the detection performance. In this study, two different sets of selected features have been adopted to train four machine-learning based classifiers. The two sets of selected features are based on Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) approach respectively. These evolutionary-based algorithms are known to be effective in solving optimization problems. The classifiers used in this study are Naïve Bayes, k-Nearest Neighbor, Decision Tree and Support Vector Machine that have been trained and tested using the NSL-KDD dataset. The performance of the abovementioned classifiers using different features values was evaluated. The experimental results indicate that the detection accuracy improves by approximately 1.55% when implemented using the PSO-based selected features than that of using GA-based selected features. The Decision Tree classifier that was trained with PSO-based selected features outperformed other classifiers with accuracy, precision, recall, and f-score result of 99.38%, 99.36%, 99.32%, and 99.34% respectively. The results show that using optimal features coupling with a good classifier in a detection system able to reduce the classifier model building time, reduce the computational burden to analyze data, and consequently attain high detection rate.

Key words: Intrusion detection system, Machine learning classifiers, Performance evaluation, Selected features,

Introduction:

Intrusion detection system (IDS) is one of the protection methods against network attacks and threats in most organizations in addition to firewalls, authentication and encryption. IDS model was first proposed by (1), that is a software to monitor and detect any intrusion in a system or network. A modern effective network-based IDS should be able to automate the network surveillance, analysis process and attacks detection or classification with high accuracy percentage in short amount of time (2, 3). An IDS can be categorized into signature-based, anomaly-based or hybrid-based. Signature-based IDS only accurately detects known attacks while anomaly-based IDS able to detect unknown attacks by comparing the current profiles against the predefined normal behaviours. The later method is effective against zero-day attacks, but it still has high false positive rates (4, 5) and hence of recent, hybrid method has been developed to overcome these limitations (6).

Due to the privacy and security issues, getting a reasonably large and complete real-world network traffic data with attacks footprints for IDS performance assessment has been made difficult. Alternatively, researchers use the publicly available benchmark datasets, namely KDD CUP 99 and NSL-KDD to evaluate the IDS performance. The NSL-KDD dataset has been used extensively, including in this work, as it provides an improved version of the original KDD Cup 99 dataset that contains huge amount of redundant records (7). Nonetheless, NSK-KDD still consists of large network traffic volumes with 125,973 instances of 41 network-related features and an assigned label to classify each record instance as either normal or abnormal. Analyzing a huge dataset imposes a heavy computational burden and hence increases the processing time. Feature selection or reduction approach has then been proposed to solve such problem. Feature selection identifies and removes irrelevant features that do not contribute to the accuracy of a predictive model and has been widely used in machine learning, data mining and data analysis (8). Using reduced set of features, also known as the selected features, it reduces the complexity of the developed model, that is reduces the building classifier model time (9).

This study investigates the performance of an IDS that uses only few selected features, as opposed to all 41 features using popular machine-learning based classifiers. Different features that have been selected using the evolutionary-based feature selection techniques from another research work have been adopted. In specific, 11 features selected using Genetic Algorithm (GA) by (10) and 20 features selected using Particle Swarm Optimization (PSO) by (11) have been used and hence, the feature selection implementation is not within the scope of this study. Machine learning (ML) techniques have been widely used for network intrusion detection as they are able to classify benign and attack patterns precisely. ML algorithms automate the improvement of their detection accuracy with subsequent trainings which may contain new and previously unseen data. However, building ML models are time consuming with the increase of data volumes (12). Hence, reducing the volumes of data to be processed using feature reduction method is critical to improve the detection performance. In this work, four state-of-the-art machine learning classifiers, namely Naïve Bayes, k-Nearest Neighbor, Decision Tree and Support Vector Machine have been implemented and evaluated. The detection accuracy of the abovementioned classifiers using different sets of features values were studied.

This paper is structured as follows. Section 2 presents an overview of the machine learning-based classifiers used in this. Section 3 discusses some of the IDS models using different machine learning classifiers. Existing feature selection approaches are covered in Section 4. Section 5 reviews some of the related works on IDS models using different feature selection methods and classifiers. Section 6 presents the experimental setup including the dataset and performance metrics used in this study. The performance of classifier models using different sets of selected features are compared and discussed in Section 7. Final comments and conclusions are provided in Section 8.

Machine Learning Classifiers:

Machine learning (ML) enables the IDSes to detect new attacks without human intervene. ML allows the IDS to change its execution strategy based on the recently acquired data. In general, there are two types of learning techniques namely the supervised and unsupervised learning. Supervised learning involves algorithms that are ‘taught’ by examples, with the input and out-put labels are provided during training (13). The unsupervised learning algorithms are left to interpret the data without guidance as no labeled data are provided in training dataset. Unsupervised learning identifies similarities and differences in data by clustering and association techniques (14).

The machine learning-based classifiers used in this study are the supervised probabilistic-based Naïve Bayes, k-Nearest Neighbors, Decision Tree and Support Vector Machine. All these classifiers are part of the state-of-the-art classifiers for they have been widely used for classification and regression problems due to their effectiveness. The theoretical background of these algorithms has been heavily discussed in many published works and hence not discuss in depth in the following subsections. The following subsections discuss the classifiers in general including their historical backgrounds, recent development and applications.

Naïve Bayes

Naïve Bayes (NB) classifier is a probabilistic-based classifier which uses Bayes’ theorem and assumes features are independent of each other and their weight are equally important (15). One of NB problems is the ‘Zero frequency or probability’ situation in which the model is not able to make prediction if it has not observed a certain category in the training data set, yet a new and unseen-before input variable appears in the test data set. Smoothing techniques such as Laplace estimation can be applied to avoid this undesirable situation (16).

With some improvements made towards the traditional NB, it has been used extensively in text classification area, along with other classification areas as it is simple to implement, computationally fast and robust (17, 18). Moreover, Naïve Bayes are among the simplest Bayesian network models that can achieve higher accuracy level if coupled with kernel density estimation (19, 20).

k-Nearest Neighbors

The k-Nearest Neighbors (kNN) is a non-parametric classification method that has been widely used due to its simplicity and effectiveness (21). kNN was first described by Fix and Hodges in 1951 (22) in a USAF School of Aviation Medicine technical report and later expanded by Cover and Hart (23). kNN classifies each unlabeled data, *t* based on the k nearest neighbors, known as the *neighborhood of t*. Majority voting among the data label in the neighborhood is then used to decide the classification for *t* with or without consideration of distanced-based weighting.

kNN requires no prior knowledge on the distribution of the data (24). However, kNN is biased by the selection of the k value. One way in choosing good k value is to run the algorithm many times and choose the one with the best performance. One of the disadvantages of this classifier is its computational cost is considerably high as it needs to compute distance the unlabeled data *t* to all training samples. One promising approach made to improve the kNN accuracy is by clustering technique (25, 26). kNN has been deployed in many domain areas including text mining, agriculture and medicine but has been heavily applied in finance-related areas such the stock market forecasting, bank customer profiling, managing financial risk as well as money laundering analyses (27).

Decision Tree

Decision Tree (DT) is a supervised learning method that maps from observations about a data to conclusions about its target value (28). The leaves represent the class or the label, the non-leaf nodes are the features and the branches represent conjunction of features that lead the specific a class. To create a DT, the training data or records are distributed recursively according to the attribute values (29).

DT is computationally fast even when dealing with large training sets since they are generally balanced and hence traversing the tree from root to the leaf requires approximately O(log2 N). The tree-based algorithms include ID3 (Iterative Dichotomiser 3), C4.5 (successor of ID3), CART (Classification and Regression Tree), CHAID (Chi-Square Automatic Interaction Detection), MARS (Multivariate Adaptive Regression Splines) and cTree (Conditional Inference Trees). One of the main challenges in DT is to build a good decision tree, that is smallest decision tree possible. Nonetheless, DT is one of the most used techniques in IDS for its fast adaptation, simplicity, and accuracy (30).

Support Vector Machine

A Support Vector Machine (SVM) is based on statistical learning theory and was developed by Vapnik in 1995 (31). SVM finds the optimal hyperplane that differentiates any two classes efficiently. By using different types of kernel functions, the low dimensional input space is transformed to a high dimensional space. Hence these nonseparable classes can then be separated by adding more dimensions. Linear, sigmoid, polynomial and radial basis functions (RBF) are some of the commonly used kernel functions, which play a significant role in SVM (32).

SVMs have performed well in multiple areas of biological analysis including analysing RNA-Sequencing and microarray gene expression data due to their capabilities to generalize well with high dimensional data (33, 34). However, SVM's performance may degrade when data is not linearly separable and having large data sets to process, as the precompute of the kernel matrix might become infeasible (35).

Intrusion Detection System using Machine Learning Classifiers:

Machine learning (ML) has been widely used in network intrusion detection for its ability to classify benign and attack patterns with high precision. Table 1 presents the performance evaluation of IDS models with different ML classifiers.

Table 1. IDS Models with Different Machine Learning Classifiers.

Authors / Year	Dataset	Classification	Accuracy	Precision	Recall	F-Score
(Relan & Patil, 2015) (36)	NSL_KDD	Decision Tree	**93.82%**			
		Naive Bayes	81.66%			
		Random Forest	92.79%			
		Multi-layer Perceptron	92.26%			
		SVM	65.01%			
(Belavagi & Muniyal, 2016) (37)	NSL_KDD	Logistic Regression	84%	83%	85%	82%
		SVM	75%	76%	79%	77%
		Naive Bayes	79%	79%	81%	78%
		Random Forest	**99%**	**99%**	**99%**	**99%**
(Amira et al., 2017) (38)	NSL_KDD	Naïve Bayes		84.41%	78.51%	81.35%
		BFTree		**98.19%**	68.32%	80.58%
		J48		**98.59%**	65.52%	78.72%
		Multi-layer Perceptron		98.24%	62.51%	76.41%
		NBTree		**98.36%**	62.64%	76.53%
		Random Forest		**98.61%**	62.58%	76.57%
(Suleiman & Isaac, 2018) (39)	NSL_KDD	Random Forest	**99.76%**	**99.9%**	99.6%	99.7%
		Decision Tree (J48)	**99.55%**	**99.5%**	99.5%	99.5%
		k-Nearest Neighbor	99.44%	99.5%	99.3%	99.4%
		Naive Bayes	88.59%	89.7%	87.7%	88.7%
		SVM	97.32%	98.3%	95.9%	97.1%
		ANN	98.24%	98.9%	97.3%	98.1%
(Devi & Abualkibash, 2019) (40)	NSL_KDD	Logistic Regression	79.7%			
		Decision tree	81.05%			
		k-Nearest Neighbor	94.17%			
		SVM	83.09%			
		Random Forest	**99.0%**			
		Adaboost	90.73%			
		Multi-layer Perceptron	80.5%			
		Naïve Bayes	92.4%			

A decision tree-based intrusion detection system was presented by (36) and a comparison study among the listed classifiers shows that the proposed model able to achieve high detection accuracy rate, at around 93.82%. Belavagi & Muniyal (37) presented classification and predictive models for intrusion detection by using machine learning classification algorithms namely Logistic Regression, Support Vector Machine, Naive Bayes and Random Forest (RF). Experimental results show that RF outperformed the other methods in all metrics with highest value of 99%. Amira et al. (38) implemented Naive Bayes, BFTree, Decision Tree (J48), Multilayer Perceptron, NBTree and Random Forest (RF) classifiers and compare their results. The results of the decision tree-based algorithms show high precision rate, which are above 98% while NB peformed the worst in this study. Suleiman and Isaac (39) evaluated six classifiers which are the Decision Tree (J48), Random Forest (RF), k-Nearest Neighbor (kNN), Naive Bayes (NB), Support Vector Machine (SVM) and Artificial Neural Networks (ANN). The experimental results show that RF and J48 classifiers outperformed others in accuracy and false positive rate. These tree-based classifiers managed to attain above 99% for accuracy and precision. Different machine learning algorithms been implemented by (40) that include Logistic Regression, Decision Tree (DT), Stochastic Gradient Descent (Adaboost), SVM, Random Forest(RF), Naive Bayes and Multilayer Perceptron and as expected, RF performed the best with achieved the highest accuracy rate of 99.0%. In summary, Decision Tree and Random Forest (which are composed of multiple decision trees) performed well in most studies in comparison to other classifiers for they are known to be efficient and accurate.

Feature Selection:

IDS normally handle vast amounts of data traffic containing redundant and irrelevant features, which could negatively affect its detection performance. Many studies have shown that classifier that is developed with an efficient subset of relevant features provides higher predictive accuracy compared to a classifier developed from the complete set of features (41, 42). Feature Selection (FS) is a popular preprocessing technique aims to find the most relevant features, that is features that have high correlation with the respective results (43). Using only relevant features in building the predictive model, it reduces the complexity of the developed model, hence reduces the building classifier model time and improve the accuracy and efficiency. In general, FS approaches can be classified into three categories, which are the wrapper, filter and hybrid (44). Filter methods only consider the relevance between features and class labels, independent of the classifiers as depicted in Figure 1. It ranks the features using statistical techniques such as t-test or fisher discriminant ratio, information theory, correlation coefficient, variance threshold as well as using distance measurement (45). These methods require less computational resources and faster than wrapper methods as no cross-validation process is performed.

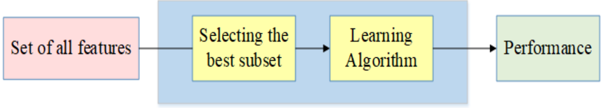


Figure 1. Filter Methods(Khalid etal., 2017)(46)

In wrapper methods, the incremental learning sessions from the specific machine learning algorithm is integrated into the feature selection process as depicted in Figure 2. The prediction performance of the algorithm is tested using different feature subsets and finally, the subset with the best performance is selected. Wrapper methods which are based on greedy search algorithms generally achieve high accuracy than filter methods. Wrapper methods for feature selection can be categorized into step forward feature selection, step backwards feature selection and exhaustive feature selection. Meanwhile, the hybrid methods, also known as embedded methods combine both filtering and wrapping methods to obtain the best of both techniques.

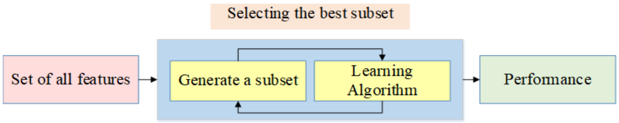


Figure 2. Wrapper Method. (Nolan & Lally, 2018)(47).

Related Works:

The following paragraphs discuss some of the existing IDS models with various feature selection techniques and classifiers and Table 2 presents the summarized information. Sarvari et al. (10) proposed an intrusion detection system using a hybrid SVM approach with Genetic Algorithm (GA) FS method. GA is a stochastic optimization algorithm, that is based on natural evolution aims to find the optimal solution. Hence, by implementing GA, the number of important features has been reduced from 41 to 11. These 11 significant features are categorized into three groups, ranked based on their importance. The 4 most important features are placed in the first priority, 5 features in the second priority and 2 least important featured in the third. The results demonstrate that the proposed algorithms, GA and SVM can attain true positive and false positive values of 97.3% and 0.17% respectively.

Ambusaidi et al. (48) proposed a common information-based algorithm that choose the ideal element for grouping. This new filter-based feature selection method is an enhancement of Mutual Information Feature Selection (MIFS) and Modified Mutual Information-based Feature Selection (MMIFS) known as Flexible Mutual Information Feature (FMIFS). They employed Least Square SVM (LS-SVM) classifier to detect the attacks in NSL-KDD dataset with their proposed system is known as LSSVM-IDS-FMIFS. The FMIFS selected 17 most significant features, that are columns 1, 2, 3, 4, 8, 10, 11, 12, 19, 23, 24, 25, 29, 31, 32, 36, and 39. The proposed system achieved 99.94% in accuracy, 98.93% in detection rate, and 0.28% of false positive rate.

Thaseen and Kumar (49) have proposed an intrusion detection model that uses rank-based chi-square feature selection technique and multi class SVM classifier. Chi-squared is a numerical test that measures deviation from the expected distribution considering the feature event is independent of the class value. Multi-class SVM is used to classify the different types of attacks in the NSL-KDD dataset. Using the proposed model 31 features were selected out of 41. The proposed system achieved 98% in accuracy and 0.13% false positive rate.

Chakir et al. (11) improved IDS efficiency by using the Information Gain (IG) feature selection method and SVM with Particle Swarm Optimization (PSO) for improved classification. PSO is a stochastic approach that performs searches using population or swarm of particles. Experiments were performed on the dataset NSL KDD and top ranked 20 features were selected. The experimental studies indicate that the proposed IG-PSO-SVM detection model performed well with 0.9% false alarm rate and 99.8% accuracy as well as precision.

Al-Yaseen (41) suggested a wrapper feature selection method, based on the firefly algorithm and SVM. The SVM model was used to assess each of the subsets of features selected from the firefly approach. The key benefit of the proposed system is its ability to adjust the firefly algorithm to match the selection of features and 10 top ranked features are selected. Their solution achieved about 78.89% in accuracy, and only 75.81% when uses all 41 features. The results of the analysis show the effectiveness of proposed feature selection technique in improving the detection system.

Table 2. IDS Models with Feature Selection and Classifiers.

Authors, Year	**Proposed IDS Models with Different Feature Selection Approaches and Classifiers**	**Dataset (No. of Features)**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**
Sarvari et al., 2015 (10)	Genetic Algorithm (GA) FS and Support Vector Machine (SVM)	KDD Cup 99 (11 features)	n/a	97.2	97.3
Ambusaidi et al., 2016 (48)	Flexible Mutual Information FS and Least Square Support Vector Machine (LSSVM)	NSL-KDD (17 features)	99.94	n/a	n/a
Thaseen and Kumar, 2017 (49)	Chi-square FS and multi class SVM	NSL-KDD (31 features)	98	n/a	n/a
Chakir et al., 2018 (11)	Information Gain (IG) FS with Particle Swarm Optimization (PSO) and SVM	NSL-KDD (20 features)	99.8	n/a	99.8
Al-Yaseen, 2019 (41)	Firefly Algorithm (FA) FS and SVM	NSL-KDD (10 features)	78.89	n/a	n/a

This study investigates the performance of the evolutionary-based feature selection methods when coupled with some of the state-of-the-art classifiers in detecting attacks in the NSL KDD data set. Therefore, the 11 GA-based selected features and the 20 PSO-based selected features by Sarvari et al. (10) and Chakir et al. (11) respectively have been adopted in this work. As mentioned earlier, implementing evolutionary-based feature selections is not within the scope of the study. The authors use the features that have been selected from the abovementioned works and evaluate the performance of these two approaches. The following paragraph provides some background on the evolutionary computing that has gained increasing attention from researchers.

Due to the optimization capabilities of the evolutionary-based feature selection techniques, these algorithms have gained much attention from the researchers. Among the popular algorithms include Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and ant colony optimization that have been widely used (50-52). Genetic algorithms are randomized search algorithms that rely on biologically inspired operators such as mutation, crossover, selection and reproduction to provide optimization. GA is an iterative process that evolves in time and using the rule of survival of the fittest to arrive at the best solution. It operates on string structures like biological structures and in every generation, a new set of strings is created using parts of the fittest members of the old set. GA is computationally costly and can take a long time to converge due to its stochastic nature (53). PSO was inspired by the movement behavior exhibits by the flocks of birds and swarms of insects. Proposed by Elberhart and Kennedy (54). PSO consists of individuals or known as particles that have a position and a velocity. Using a mathematical formula, it iteratively improves the solution by moving these particles in the given search-space. The movement of each particle is influenced by its local best-known position but is also guided toward the best-known positions in the search-space, updated by other particles that have found better positions. This moves the swarm toward the best solutions.

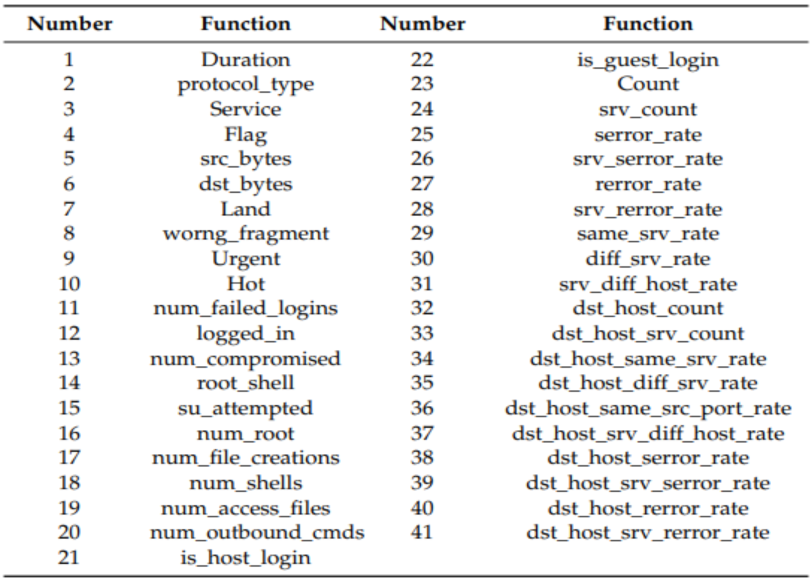
PSO is easy to implement and computationally inexpensive compared to GA. However, with more features in the data set, the solution space increases rapidly. In addition, high number of uncorrelated or redundant features result in many local optima detected in a large solution space and thus, evolutionary-based methods still suffer from the local optimal stagnation problems (45). In this work the data dimension is limited to 41 and hypothetically, PSO should be able to converge fast and expected to have less selected features than GA. However, based on Table 2, the selected features of PSO derived by (11) is higher than those of GA derived by (10). This could due to the selection of Information Gain threshold value used in the experiments that led to 20 important features been selected. Similarly, another work that deploys a hybrid model that integrates Gini Index with PSO can be found in (55). The authors only consider features as important thus selected when the respective Gini Index's scores are less than 0.4. Consequently, only 18 features are selected from the NSL-KDD dataset in their work. In this study, two sets of selected features, one with 11 features selected using GA and another set of 20 features selected using PSO, have been adopted to train the four different predictive models.

Methodology:

The NSL-KDD dataset, proposed intrusion detection system and performance metrics used in this study are discussed in the following subsections.

Dataset

NSL-KDD dataset (56), is an improved version of KDD-CUP 99 dataset that has been used in this study. It has no redundant and duplicate records and thus, better detection rate is expected. In this dataset, there are 125,973 instances with 41 attributes or features and one assigned label to indicate the record as normal or abnormal. Figure 3 depicts the 41 features of the NSL-KDD dataset. These features can be divided into three different categories as follows: 1) features extracted from the TCP/IP connection, 2) features to access TCP packet payload and 3) time-based traffic features and host-based traffic features. The attacks in this dataset can be classified into four different types of attacks, namely the DoS, Probe, U2R and R2L attacks. This public benchmark dataset has been widely used by many researchers to conduct different types of analyses and develop effective IDSes (57- 60).



Number	Function	Number	Function
1	Duration	22	is_guest_login
2	protocol_type	23	Count
3	Service	24	srv_count
4	Flag	25	serror_rate
5	src_bytes	26	srv_serror_rate
6	dst_bytes	27	rerror_rate
7	Land	28	srv_rerror_rate
8	worng_fragment	29	same_srv_rate
9	Urgent	30	diff_srv_rate
10	Hot	31	srv_diff_host_rate
11	num_failed_logins	32	dst_host_count
12	logged_in	33	dst_host_srv_count
13	num_compromised	34	dst_host_same_srv_rate
14	root_shell	35	dst_host_diff_srv_rate
15	su_attempted	36	dst_host_same_src_port_rate
16	num_root	37	dst_host_srv_diff_host_rate
17	num_file_creations	38	dst_host_serror_rate
18	num_shells	39	dst_host_srv_serror_rate
19	num_access_files	40	dst_host_rerror_rate
20	num_outbound_cmds	41	dst_host_srv_rerror_rate
21	is_host_login		

Figure 3. The 41 features of the NSL-KDD dataset (9).

Design and Implementation

Figure 4 shows the proposed IDS model used in this study and the processes involved. These processes include pre-processing data, using selected features, building classification models, and evaluating performance are then elaborated in the following subsections.

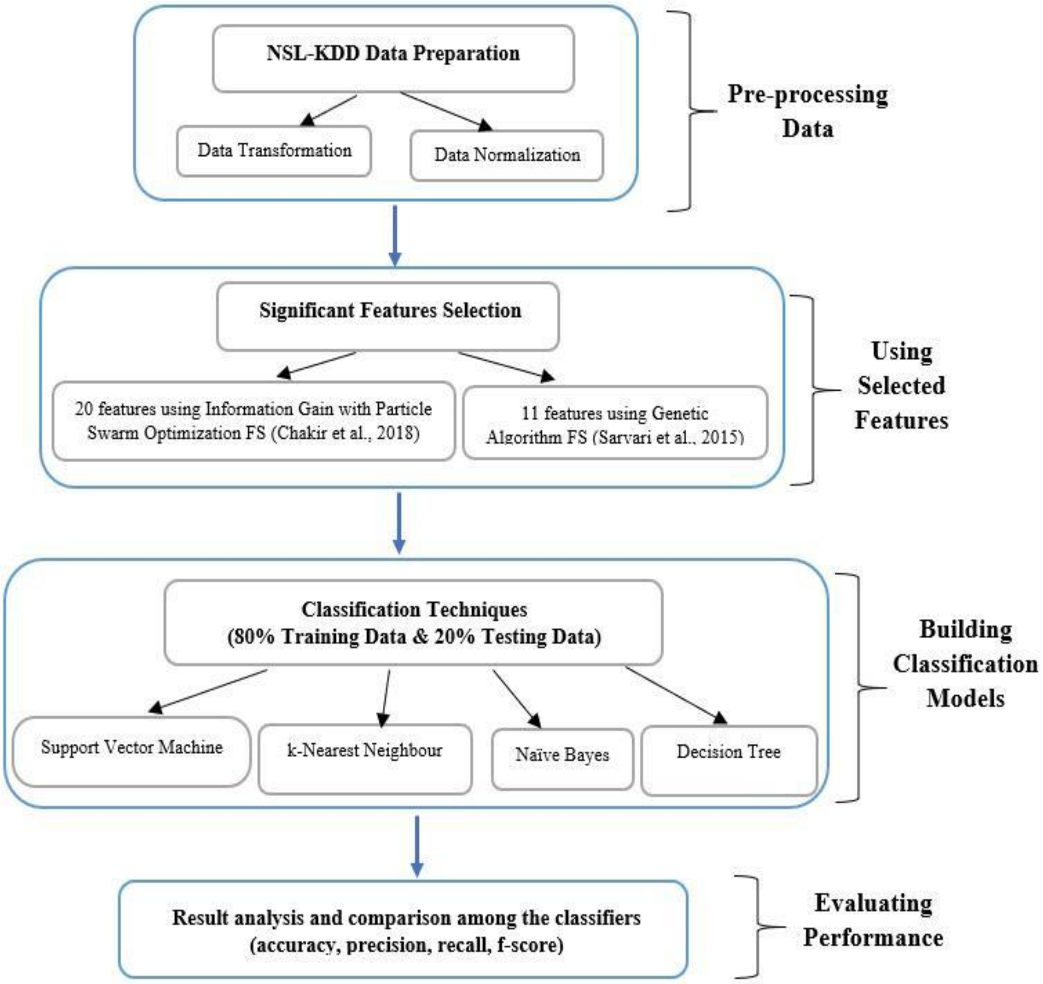
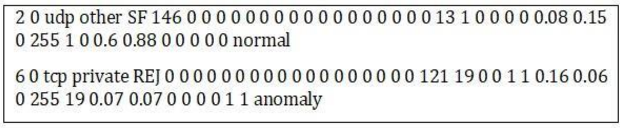

NSL-KDD Data Preparation
Data Transformation
Data Normalization
Pre-processing Data
Significant Features Selection
20 features using Information Gain with Particle Swarm Optimization FS (Chakir et al., 2018)
11 features using Genetic Algorithm FS (Sarvari et al., 2015)
Using Selected Features
Classification Techniques (80% Training Data & 20% Testing Data)
Support Vector Machine
k-Nearest Neighbour
Naïve Bayes
Decision Tree
Building Classification Models
Result analysis and comparison among the classifiers (accuracy, precision, recall, f-score)
Evaluating Performance

Figure 4. Proposed IDS Model.

Pre-processing Data: Data Transformation and Normalization

Figure 5 shows two records taken from the NSL-KDD dataset, in specific records for line 2 and 6 that contain mixed of numerical and string values. These strings or nominal feature values need to be transformed into numeric values with the affected columns are columns number 2 (Protocol_type), 3 (Services), 4 (Flag) and 42 (Attack or Normal). The data in column 42 for each record has been transformed, in particular the 'normal' value has been assigned to value 0 and the 'anomaly' value has been assigned to value 1.



```
2 0 udp other SF 146 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 13 1 0 0 0 0 0.08 0.15
0 255 1 0 0.6 0.88 0 0 0 0 0 normal

6 0 tcp private REJ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 121 19 0 0 1 1 0.16 0.06
0 255 19 0.07 0.07 0 0 0 0 1 1 anomaly
```

Figure 5. NSL-KDD Records.

Due to the large variation among some of the feature values, for example values 146 and 0.08 as shown in line 2 of Figure 5, normalization is required for better performance. Normalization scales the data features into a specific range without altering the feature's statistical properties. The maximum and minimum values of the features were determined, and data is converted into a normalized form using the following equation:

$$Normal\ data_i = \frac{(data_i{-}mindata)}{(maxdata{-}data\ )}$$

Using Selected Features: Adopting Two Sets of Selected Significant Features

In this study, two sets of selected significant features have been applied. The 20 selected features by (11) obtained using Information Gain and Particle Swarm Optimization (PSO) and 11 selected features by (10) obtained using Genetic Algorithm (GA) are fed into the machine learning models. Both GA and PSO are evolutionary algorithms with their own advantages and limitations. Table 3 shows the selected features in these two sets. Most of the features selected by GA are also selected by PSO-based feature selection approach. However, PSO-based feature selection technique considers additional 9 features are also relevant and important in classifying attacks accurately.

Table 3. Different Sets of Selected Features.

No	Selected Significant Features	
	Set 1: 20 Features	**Set 2: 11 Features**
1	src_bytes	protocol type
2	Service	Service
3	dst_bytes	Flag
4	Flag	wrong fragment
5	diff_srv_rate	logged_in
6	same_srv_rate	numfile creations
7	dst_host_srv_count	Count
8	dst_host_same_srv_rate	dst_host_same_srv_rate
9	dst_host_diff_srv_rate	dst_host_same_srv_port_rate
10	dst_host_serror_rate	is_gus login
11	logged_in	srv_diff_host_rate
12	dst_host_srv_serror_rate	
13	serror_rate	
14	Count	
15	srv_serror_rate	
16	dst_host_srv_diff_host_rate	
17	dst_host_count	
18	dst_host_same_src_port	
19	srv_diff_host_rate	
20	srv_count	

Building Classification Models: Training/Testing Data and Predictive Models

The NSL-KDD data are split into training and testing sets for supervised learning. Following the previous works by Sarvari et al. (10) and Chakir et al. (11), 80% of the data has been randomly selected and used to train the machine learning models and the rest of 20% is used for the classifier's performance evaluation. Table 4 shows the statistics of the data used in this study.

Table 4. Statistics of the NSL-KDD Dataset.

	Total	**Training (80%)**	**Testing (20%)**
Normal	67,244	53,795	13,449
Attack	58,730	46,984	11,746
Total	**125,974**	**100,779**	**25,195**

The Decision Tree, Naïve Bayes, Support Vector Machine, and k-Nearest Neighbor algorithms are implemented using MATLAB version R2018b. Using the training data, four predictive models are then built and to be used for classifying the remaining 20% of the dataset.

Performance Metrics Evaluation

The accuracy, precision, recall and F-score performance measurements are used to evaluate the performance of the classifiers with different sets of selected features. The confusion matrix is the basis for calculating the abovementioned performance metrics of the classifiers. It includes true positive (TP) that specifies the normal instances that are correctly predicted, true negative (TN) that indicates the abnormal instances that are identified correctly, false positive (FP) that denotes the abnormal instances that are wrongly assumed as normal and false negative (FN) that specifies the abnormal instances detected as normal. The descriptions of the performance metrics are as follows: -

(i) **Classification rate or Accuracy:** one of the most important performance measurements of a classification algorithm that shows the ability of the algorithm to accurately predict positive and negative instances, as shown in the following formula:

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

(ii) **Precision or the positive predictive value:** refers to the ratio of correctly predicted positive observations to the total predicted positive observations.

$$Precision = \frac{TP}{TP + FP}$$

(iii) **Recall known as sensitivity:** refers to the true positive rate that is determined correctly.

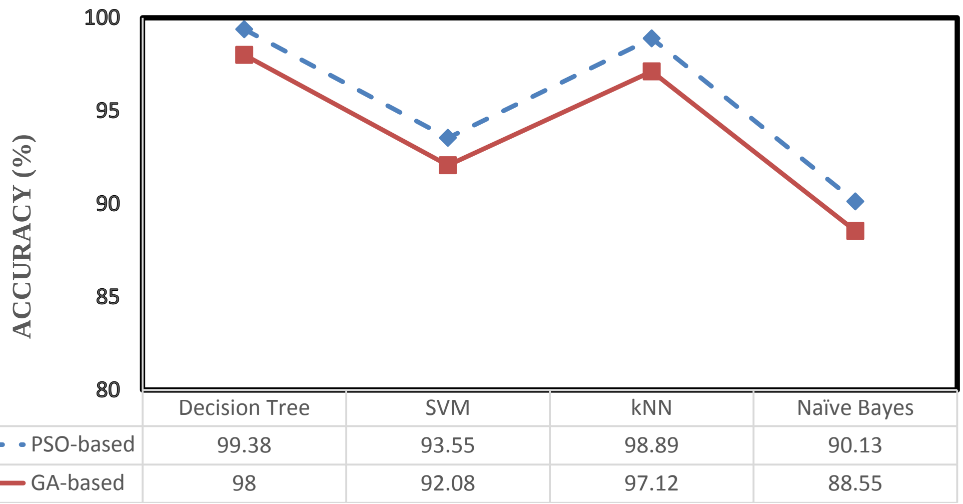
$$Recall = \frac{TP}{TP + FN}$$

(iv) **F-score:** is the harmonic mean of the precision and recall.

$$F - score = \frac{Precision * Recall}{Precision + Recall}$$

Results and Discussion:

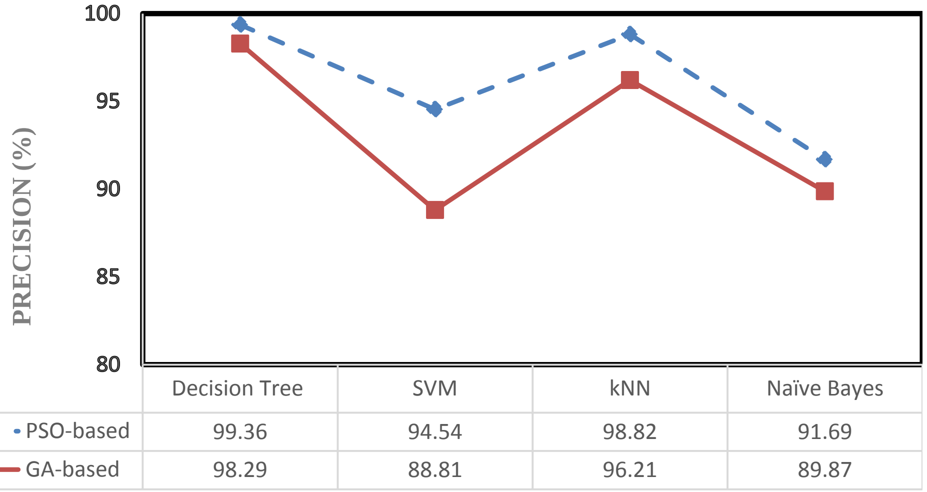
Accuracy is the most critical performance measurement in intrusion detection and Figure 6 shows all the classifiers' performances using both PSO-based and GA-based selected features sets. Interestingly, even though PSO has greater number of selected features used to develop the predictive models, the overall performance of the models is superior than that of GA's. This could due to the ability of the PSO, together with Information Gain to correctly anticipate the most relevant attack features in the dataset. In general, the accuracy improves by approximately 1.55% when implemented using the PSO-based selected features than that of using GA-based selected features. As expected, the Decision Tree (DT) classifier attained the highest accuracy percentage, which is 99.38% with PSO selected features. Meanwhile, decision tree classifier with GA-based selected features only able to detect up to 98% of accuracy. The results are consistent with the studies shown in Table 1, the decision tree's performance. In this experiment, the NB classifier performed the worst behind SVM and kNN. In summary, the classifiers' accuracy using significant features derived from PSO performed better than those with features obtained by GA.



	Decision Tree	SVM	kNN	Naïve Bayes
PSO-based	99.38	93.55	98.89	90.13
GA-based	98	92.08	97.12	88.55

Figure 6. Classifiers' Accuracy Percentage Comparison.

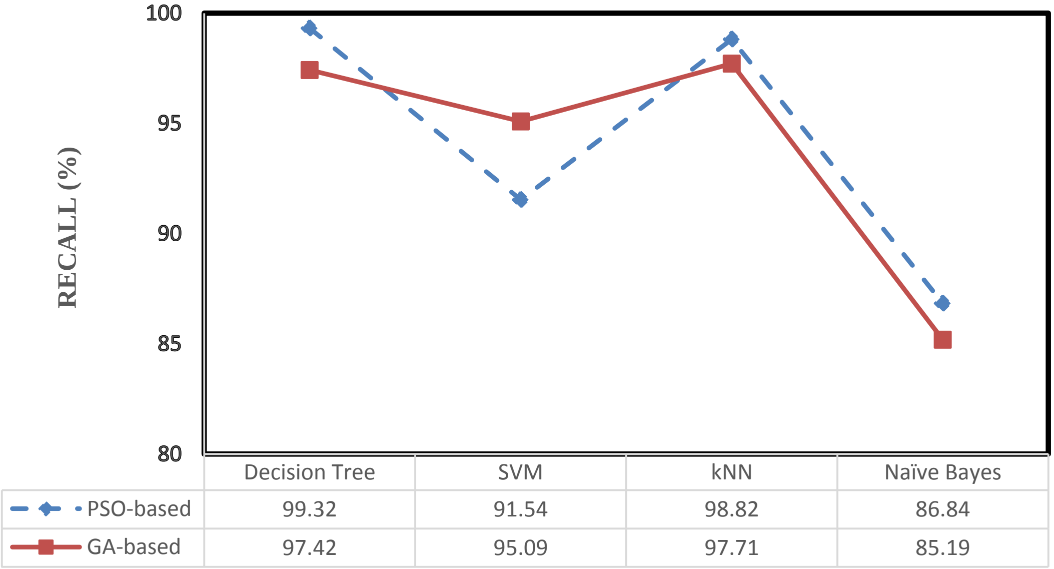
The precision results that show the classifier's percentage of predicting instances correctly is one of the important indicators of good models, are shown in Figure 7. The classifiers using PSO-based selected features outperformed the classifiers that are trained by the GA-based selected features. Again, as expected, the decision tree classifier obtained the highest precision percentages (of value 99.36%) compared to other classifiers. Unlike previous results, in this experiment, SVM has the worst precision percentage with value of 88.81%, behind NB and kNN respectively. The performance difference rate shown by SVM in these two different features sets is huge, which is about 5.73%. Meanwhile the other three classifiers are considerably consistent in their performance.



	Decision Tree	SVM	kNN	Naïve Bayes
PSO-based	99.36	94.54	98.82	91.69
GA-based	98.29	88.81	96.21	89.87

Figure 7. Classifiers' Precision Percentage Comparison.

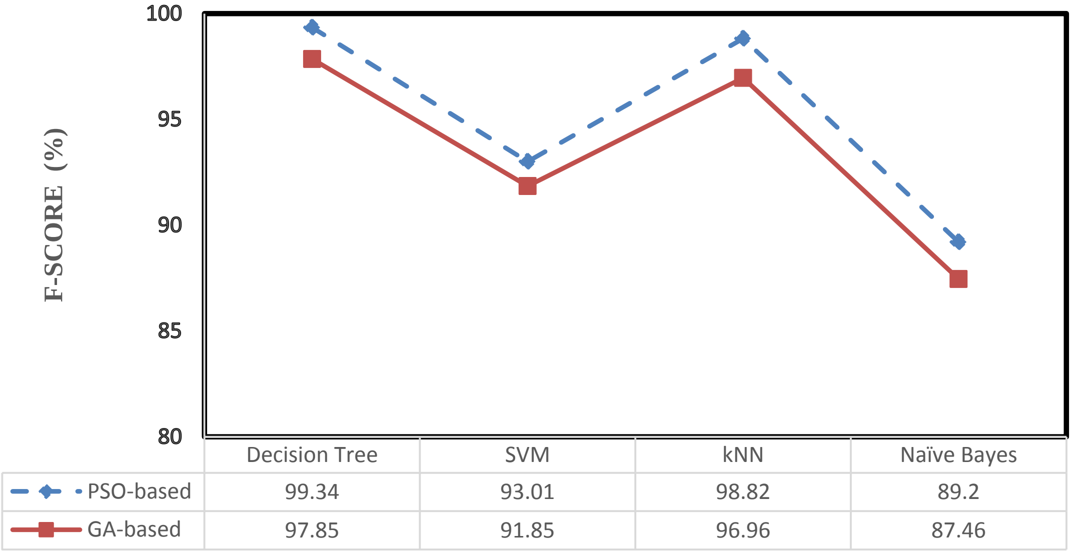
Figure 8 depicts the recall or sensitivity rate of the predictive models. The classifiers using PSO-based selected features outperformed the classifiers that are trained by the GA-based selected features except for SVM. Such problem is prominent in SVM and there has been published works discussing this phenomenon, known as the outlier sensitivity problem of standard SVM (61). Many have found SVMs do not perform well with certain noise intensities. The performance of SVM trained by the PSO-based selected features degraded with the presence of noise and even worse than that of DGA-based, by approximately 4.5%. The rest of the classifiers are consistent in their performance. The decision tree classifier again attained the highest recall percentages (of value 99.32%) compared to other classifiers.



	Decision Tree	SVM	kNN	Naïve Bayes
PSO-based	99.32	91.54	98.82	86.84
GA-based	97.42	95.09	97.71	85.19

Figure 8. Classifiers' Recall Percentage Comparison.

Figure 9 shows the f-score or f-measure rate of the predictive models. In general, the classifiers' f-score performs better by approximately 1.56% when implemented using the PSO-based selected features than that of using GA-based selected features. Decision Tree (DT) classifier attained the highest accuracy percentage, which is 99.34% with PSO selected features. In this experiment, the NB classifier performed the worst behind SVM and kNN with percentage of 87.6% using the GA-based features.



	Decision Tree	SVM	kNN	Naïve Bayes
PSO-based	99.34	93.01	98.82	89.2
GA-based	97.85	91.85	96.96	87.46

Figure 9. Classifiers' F-score Percentage Comparison.

In summary, as expected, the efficient decision tree outperformed other classifiers in all test instances, in both feature sets. The standard SVM's sensitivity rate is susceptible to noise and can be improved upon as suggested in (62). kNN's performances are also considerably good in comparison to the other two classifiers, and meanwhile the NB classifier performed the worst in most of the test.

Conclusion:

The performance of four supervised classifiers with different selected feature values on the NSL-KDD dataset were evaluated. The feature values were derived from Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) feature selection approach respectively. The experimental results show that using smaller number of selected and relevant features may not necessarily improves the accuracy. Instead, using the appropriate number of relevant and significant features, even if the number is big, it could enhance the performance of the machine learning models. The 20 features selected by PSO outperformed the 11 features selected by GA in every performance metric except for recall due to existing SVM’s outlier sensitivity problem. The adopted PSO feature selection method with Information Gain selected the top 20 relevant features from the 41 features in NSL-KDD dataset and hence improves the complexity, time, and the accuracy of the predictive models. Decision Tree has proven to be an efficient classifier and outperformed Naïve Bayes, k-Nearest Neighbor and Support Vector Machine classifiers in every evaluation test. In this experimental study, a maximum accuracy of 99.38% and precision of 99.36% have been attained by the decision tree-based IDS using particle swarm optimization feature selection. In summary, combining a good feature selection with an efficient classifier in a detection system able to reduce to complexity of data analysis and consequently improve the detection performance.

Acknowledgment:

Work by the authors was partially supported by the Universiti Putra Malaysia under the Grant Putra Scheme: GP/2018/9621500.

Authors' declaration:

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are ours. Besides, the Figures and images, which are not ours, have been given the permission for re-publication attached with the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee in Universiti Putra Malaysia.

References:

1. Denning DE. An Intrusion-Detection Model. IEEE Transactions on Software Engineering. 1987;2,222–232.
2. Axelsson S. Intrusion Detection Systems: A Survey and Taxonomy. International Journal of Innovative Technology and Exploring Engineering. 2000;99, 1–15. https://doi.org/10.1.1.1.6603
3. Azeez NA, Bada TM, Misra S, Adewumi A, Van der Vyver C, Ahuja R. Intrusion Detection and Prevention Systems: An Updated Review. Advances in Intelligent Systems and Computing. 2020;1042, 685–696. https://doi.org/10.1007/978-981-32-9949-8_48
4. Debar H. An introduction to intrusion-detection systems. Proceedings of Connect. 2000;1-18.
5. Scarfone K, Mell P. Guide to intrusion detection and prevention systems (idps). NIST Spec Publ. 2007;800,94.
6. Khraisat A, Gondal I, Vamplew P, Kamruzzaman J, Alazab A. Hybrid Intrusion Detection System Based on the Stacking Ensemble of C5 Decision Tree Classifier and One Class Support Vector Machine. Electronics. 2020;9(1),173.
7. NSL-KDD Dataset for Network-Based Intrusion Detection Systems. 2020. Available online: https://www.unb.ca/cic/datasets/nsl.html (accessed on 10 February 2020)
8. Malhotra P, Sharma P. Intrusion detection using machine learning and feature selection. Int. J. Comput. Netw. Inf. Secur. 2019;4, 43–52.
9. Alabdulwahab S, Moon B. Feature Selection Methods Simultaneously Improve the Detection Accuracy and Model Building Time of Machine Learning Classifiers. Symmetry. 2020;12(9), 1424.
10. Sarvari S, Muda Z, Ahmad I, Barati M. GA and SVM Algorithms for Selection of Hybrid Feature in Intrusion Detection Systems. Intl. Review on Computers and Software (IRECOS). 2015;10(3), 265–270.
11. Chakir EM, Moughit M, Khamlichi YI. An effective intrusion detection model based on svm with feature selection and parameters optimization. Journal of Theo-retical and Applied Information Technology. 2018;96(12), 3873–3885.
12. Liu S, Wang X, Liu M, Zhu J. Towards better analysis of machine learning models: A visual analytics perspective, Visual Informatics. 2017;1(1),48-56.
13. Dhanda N, Datta SS, Dhanda M. Machine Learning Algorithms. Journal of Communications and Information Networks. 2019;210–233. https://doi.org/10.4018/978-1-5225-7955-7.ch009
14. Amruthnath N, Gupta T. A Research Study on Unsupervised Machine Learning Algorithms for Early Fault Detection in Predictive Maintenance. Computers and Electrical Engineering. 2018;355–361. https://doi.org/10.13140/RG.2.2.28822.24648
15. Lewis D. Naive Bayes at forty: the independence assumption in information retriev-al. In Machine Learning: ECML-98, Proceedings of the 10th

European Conference on Machine Learning, Chemnitz, Germany. 1998;4–15.

16. He F, Ding X. Improving Naive Bayes Text Classifier Using Smoothing Methods. In: Amati G, Carpineto C, Romano G. (eds) Advances in Information Retrieval. ECIR 2007. Lecture Notes in Computer Science. 2007;4425. Springer. https://doi.org/10.1007/978-3-540-71496-5_73
17. Granik M, Mesyura V. Fake news detection using naive Bayes classifier. IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), Kiev. 2017;900-903.
18. Xu S. Bayesian Naïve Bayes classifiers to text classification. Journal of Information Science. 2018;44(1), 48-59.
19. Sasongko TB, Arifin O, Al Fatta H. Optimization of Hyper Parameter Band-width on Naïve Bayes Kernel Density Estimation for the Breast Cancer Classification. 2019 International Conference on Information and Communications Technology (ICOIACT), Yogyakarta. 2019;226-231.
20. Murakami Y. Mizuguchi K. Applying the Naive Bayes classifier with kernel density estimation to the prediction of protein-protein interaction sites. Bioinformatics. 2010;26, 1841-8.
21. Hand D, Mannila H, Smyth P. Principles of Data Mining:MIT Press, Cambridge. 2001.
22. Fix E, Hodges JL. Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties (Report). USAF School of Aviation Medicine, Randolph Field, Texas. 1951.
23. Cover T, Hart P. Nearest neighbor pattern classification. IEEE Transactions on Information Theory. 1967;13(1),21-27.
24. Dasarathy BV. Nearest Neighbor (NN) Norms NN Pattern Classification Techniques. IEEE Computer Society Press, Los Alamitos. 1991.
25. Alizadeh H, Minaei B, Kasmani, AK Saeed. A New Method for Improving the Per-formance of K Nearest Neighbor using Clustering Technique. JCIT. 2009;4, 84-92.
26. Jiang S, Pang G, Wu M, Kuang L. An improved K-nearest-neighbor algorithm for text categorization, Expert Systems with Applications. 2012;39(1), 1503-1509. https://doi.org/10.1016/j.eswa.2011.08.040
27. Imandoust SB, Bolandraftar M. Application of K-nearest neighbor (KNN) approach for predicting economic events theoretical background. Int J Eng Res Appl. 2013;3, 605-610.
28. Quinlan JR. Induction of decision trees. Machine Learning. 1986;1, 81–106
29. Safavian SR, Landgrebe DA. Survey of Decision Tree Classifier Methodology. IEEE Transactions on Systems, Man and Cybernetics. 1991;21, 660-674.
30. Sani HM, Lei C, Neagu D. Computational Complexity Analysis of Decision Tree Algorithms. In: Bramer M., Petridis M. (eds) Artificial Intelligence XXXV. SGAI 2018. Lecture Notes in Computer Science:Springer 2018;11311. https://doi.org/10.1007/978-3-030-04191-5_17
31. Cortes C, Vapnik V. Support-vector networks. Mach Learn. 1995;20, 273–297. https://doi.org/10.1007/BF00994018
32. Burges CJ. A tutorial on support vector machines for pattern recognition. Data Mini. Knowl. Discov. 1998;2, 121–167.
33. Huynh P, Nguyen V, Do T. Novel hybrid DCNN–SVM model for classifying RNA-sequencing gene expression data, Journal of Information and Telecommunication. 2019;3(4), 533-547.
34. Li Z, Xie W, Liu T. Efficient feature selection and classification for microarray data. PLoS ONE. 2018;13(8), e0202167. https://doi.org/10.1371/journal.pone.0202167
35. Cervantes J, Garcia-Lamont F, Rodríguez-Mazahua L, Lopez A. A comprehensive survey on support vector machine classification: Applications, challenges and trends, Neurocomputing. 2020;408,189-215. https://doi.org/10.1016/j.neucom.2019.10.118
36. Relan NG, Patil DR. Implementation of network intrusion detection system using variant of decision tree algorithm. International Conference on Nascent Technologies in the Engineering Field. 2015;1–5. https://doi.org/10.1109/ICNTE.2015.7029925
37. Belavagi MC, Muniyal B. Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection. Procedia Computer Science. 2016;89, 117–123. https://doi.org/10.1016/j.procs.2016.06.016
38. Amira AS, Hanafi SEO, Hassanien AE. Comparison of classification techniques applied for network intrusion detection and classification. Journal of Applied Logic. 2017;24, 109–118. https://doi.org/10.1016/j.jal.2016.11.018
39. Suleiman MF, Issac B. Performance comparison of intrusion detection machine learning classifiers on benchmark and new datasets, 28th International Conference on Computer Theory and Applications (ICCTA 2018), Alexandria. 2018.
40. Devi RR, Abualkibash M. Intrusion Detection System Classification Using Different Machine Learning Algorithms on KDD-99 and NSL-KDD Datasets - A Review Paper. International Journal of Computer Science and Information Technology. 2019;11(03), 65–80. https://doi.org/10.5121/ijcsit.2019.11306
41. Al-Yaseen WL. Improving intrusion detection system by developing feature selec-tion model based on firefly algorithm and support vector machine. IAENG Interna-tional Journal of Computer Science, 2019;46(4), 1–7.
42. Najeeb RF, Dhannoon BN. A feature selection approach using binary Firefly Algorithm for network intrusion detection system. ARPN Journal of Engineering and Applied Sciences, 2018;13(6), 2347–2352.
43. Dash M, Liu H. Feature Selection for Classification. Intelligent Data Analysis. 1997;1(3), 131–156.
44. Miao J, Niu L. A Survey on Feature Selection. Procedia Computer Science, 91 (Itqm). 2017;919–926. https://doi.org/10.1016/j.procs.2016.07.111

45. Xue B, Zhang M, Browne WN. Particle swarm optimisation for feature selection in classification: novel initialisation and updating mechanisms. Applied Soft Computing. 2014;18, 261–276.
46. Khalid S, Khalil T, Nasreen S. A survey of feature selection and feature extraction techniques in machine learning. Procedia Computer Science. 2017;372–378. https://doi.org/10.1109/SAI.2014.6918213
47. Nolan DR, Lally C. Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. Journal of Computational Science. 2018;24,132–142. https://doi.org/10.1016/j.jocs.2017.04.009
48. Ambusaidi MA, He X, Nanda P, Tan Z. Building an intrusion detection system using a filter-based feature selection algorithm. IEEE Transactions on Computers. 2016;65(10),2986–2998. https://doi.org/10.1109/TC.2016.2519914
49. Thaseen IS, Kumar CA. Intrusion detection model using fusion of chi-square feature selection and multi class SVM. Journal of King Saud University - Computer and Information Sciences. 2017;29(4),462–472. https://doi.org/10.1016/j.jksuci.2015.12.004
50. Aghdam MH, Ghasem-Aghaee N, Basiri ME. Text Feature Selection Using Ant Colony Optimization, Expert Systems with Applications. 2009;36(3), 6843-6853. https://doi.org/10.1016/j.eswa.2008.08.022
51. Aslahi-Shahri BM, Rahmani R, Chizari M. et al. A hybrid method consisting of GA and SVM for intrusion detection system. Neural Computing and Applications. 2016;27(6),1669–1676.
52. Zhang Y, Gong D, Hu Y, Zhang W. Feature selection algorithm based on bare bones particle swarm optimization. Neurocomputing. 2015;148, 150–157.
53. Xue Y, Jia W, Zhao X, Pang W, Meng W. An Evolutionary Computation Based Feature Selection Method for Intrusion Detection. Sec. and Commun. Netw. 2018. https://doi.org/10.1155/2018/2492956
54. Kennedy J, Eberhart R. Particle swarm optimization. Proc Neural Networks. Proceedings of IEEE International Conference, 1944. 1995;1942e8.
55. Li L, Yu Y, Bai S, Cheng J, Chen X. Towards Effective Network Intrusion Detection: A Hybrid Model Integrating Gini Index and GBDT with PSO. Journal of Sensor. 2018;9. https://doi.org/10.1155/2018/1578314
56. Tavallaee M, Bagheri E, Lu W, Ghorbani A. A Detailed Analysis of the KDD CUP 99 Data Set. Proceeding of the IEEE Symposium on Computational Intel-ligence for Security and Defense Applications (CISDA 2009). 2009.
57. Ding Y, & Zhai Y. Intrusion Detection System for NSL-KDD Dataset Using Convolutional Neural Networks. Proceedings of the 2018 2nd In-ternational Conference on Computer Science and Artificial Intelligence (CSAI '18), New York. 2018;81–85. https://doi.org/10.1145/3297156.3297230
58. Ingre B, Yadav A. Performance analysis of NSL-KDD dataset using ANN. Proceedings of the IEEE International Conference on Signal Processing and Communication Engineering Systems, Guntur. 2015;92–96.
59. Su T, Sun H, Zhu J, Wang S, Li Y. (). BAT: Deep Learning Methods on Network Intrusion Detection Using NSL-KDD Dataset:IEEE Access. 2020;8,29575-29585.
60. Yu Y, Bian N. An Intrusion Detection Method Using Few-Shot Learning. IEEE Access, 8, 49730-49740.
61. Atla A, Tada R, Sheng V, Singireddy N. Sensitivity of different machine learning algorithms to noise. J. Comput. Sci. Coll. 2011;2020;26(5),96–103.
62. Veropoulos K, Campbell C, Cristianini N. Controlling the Sensitivity of Support Vector Machines. Proceedings of International Joint Conference Artificial Intelligence. 1999.

تقييم أداء نظام كشف التسلل باستخدام الميزات ومصنفات مختارة في التعلم الالي

رجاء ازلينا رجاء محمود **عامر حسين عبدي** **مازنيدا حسين**

جامعة بوترا ماليزيا ، ماليزيا

الخلاصة

تتضمن بعض التحديات الرئيسية في تطوير نظام فعال للكشف عن التسلل المستند إلى الشبكة (IDS) تحليل أحجام حركة مرور الشبكة الكبيرة وإدراك حدود القرار بين السلوكيات العادية وغير الطبيعية. يمكن أن يؤدي نشر اختيار الميزات جنبًا إلى جنب مع المصنفات الفعالة في نظام الكشف إلى التغلب على هذه المشكلات. يجد اختيار الميزة أكثر الميزات ذات الصلة ، وبالتالي يقلل من الأبعاد والتعقيد لتحليل حركة مرور الشبكة. علاوة على ذلك ، فإن استخدام الميزات الأكثر صلة لبناء النموذج التنبئي ، يقلل من تعقيد النموذج المطور ، وبالتالي يقلل من وقت نموذج مصنف المبنى والذي يؤدي الى تحسن أداء الكشف. في هذه الدراسة ، تم اعتماد مجموعتين مختلفتين من الميزات المختارة لتدريب أربعة مصنّفات قائمة على التعلم الآلي. تعتمد مجموعتا الميزات المحددة على الخوارزمية الجينية (GA) ونهج تحسين حشد الجسيمات (PSO) على التوالي. من المعروف أن هذه الخوارزميات المستندة إلى التطور فعالة في حل مشاكل التحسين. المصنفات المستخدمة في هذه الدراسة هي Naïve Bayes و k-Nearest Neighbor و Decision Tree و Support Vector Machine التي تم تدريبها واختبارها باستخدام مجموعة بيانات NSL-KDD. تم تقييم أداء المصنفات المذكورة أعلاه باستخدام قيم خصائص مختلفة. تشير النتائج التجريبية إلى أن دقة الكشف تتحسن بنسبة 1.55٪ تقريبًا عند تنفيذها باستخدام الميزات المحددة المستندة إلى PSO مقارنة باستخدام الميزات المحددة المستندة إلى GA. تفوق مصنف شجرة القرار الذي تم تدريبه باستخدام الميزات المحددة المستندة إلى PSO على المصنفات الأخرى بدقة ودقة واستدعاء ونتائج f-Score بنسبة 99.38٪ و 99.36٪ و 99.32٪ و 99.34٪ على التوالي. أظهرت النتائج أن استخدام اقتران الميزات المثلى مع المصنف الجيد في نظام الكشف قادر على تقليل وقت بناء نموذج المصنف ، وتقليل العبء الحسابي لتحليل البيانات ، وبالتالي تحقيق معدل اكتشاف مرتفع.

الكلمات المفتاحية: نظام كشف التسلل، مصنّفات التعلم الآلي، تقييم الأداء، ميزات مختارة.