ORIGINAL ARTICLE

Journal Section

Intelligent Intrusion Detection System in Smart Grid Using Computational Intelligence and Machine Learning

Suleman Khan[1]* | **Kashif Kifayat**[1] | **Ali Kashif Bashir, SMIEEE**[2] | **Andrei Gurtov, SMIEEE**[3] | **Mehdi Hassan**[1]

[1] Air University, Islamabad, Pakistan

[2] School of Computing, Mathematics, and Digital Technology, Manchester Metropolitan University, United Kingdom

[3] Department of Computer and Information Science, Linköping University, Sweden

Correspondence
Suleman Khan, National Centre for Cyber Security (NCCS), Air University, Islamabad, Pakistan
Email: 171518@students.au.edu.pk

Funding information

Smart grid systems enhanced the capability of traditional power networks while being vulnerable to different types of cyber-attacks. These vulnerabilities could cause attackers to crash into the network breaching the integrity and confidentiality of the smart grid systems. Therefore, an intrusion detection system (IDS) becomes an important way to provide a secure and reliable services in a smart grid environment. This paper proposes a feature-based IDS for smart grid systems. The proposed system performance is evaluated in terms of accuracy, intrusion detection rate and false alarm rate. The obtained results show that the Random Forest and Neural Network classifiers have outperformed other classifiers. We have achieved a 0.5% false alarm rate on KDD99 dataset and a 0.08% false alarm rate on the NSLKDD dataset. The detection rate and the testing accuracy on average are 99% for both datasets.

KEYWORDS

IoT,Smart Grids, *Cyber Physical Systems*, Cyber Security, Energy, Edge, Machine Learning, KDD99, NSLKDD

Abbreviations: ABC, a black cat; DEF, doesn't ever fret; GHI, goes home immediately.

* Equally contributing authors.

1 | INTRODUCTION

Data-driven technologies is now applied to smart grid as a way of sustainable energy environment. This approach can be added to a cyber-physical system consisting of hardware, software and other physical gears. Smart grid supplies electricity on-demand to end-users from centralized stations and distribute to generating stations using information and communication technologies. Energy supplier companies supply electricity at low cost and also control the end-user demand for supply. In the smart grid system, one of the significant issues is security. Many vulnerabilities exist in cyber-physical systems and hackers take advantage of vulnerabilities to launch malicious attacks on power systems. Security problems usually include authentication, data protection, availability, confidentiality, honesty, energy efficiency, single-point failures to be tested, and more [1].The attackers destroy a whole range of cyberspace in modern electronic warfare. In our societies cybercrimes proliferated. Attacks, hacking, and malicious practices such as viruses, trojans, and spamming are common risks to individuals and nations. The digital networks of cellular telephony, wireless sensor networks, satellites, tactical military communications, Internet of Things, smart grids and Supervisory Control and Data Acquisition (SCADA) are everything vulnerable to that kind of electronic attack [2].A lot of work has been done on smart grid system implementation but the majority of work are not focusing on the security requirements for the smart grid systems [3, 4]. Intrusion detection system (IDS) plays an essential role in cyber-attacks on smart grid systems and secures them against attacks. The IDS are part of the network security domain and play a vital role in protecting and maintaining a secure network.IDS system is represented in figure 2.

A typical IDS system examines and analyzes network traffic to detect and analyze attacks, and also to prevent any security violations by generating alarms for network administrator. There are two major types of IDS: Host-based IDS and Network-based IDS. IDS can be further classified into Anomaly-based and Signature-based IDS systems [5, 6, 7]. Anomaly-based IDS detects attacks using previously recorded normal real-time traffic image and by comparing it with current traffic. Though, it is widely used in various IDS, it registers a large number of false-positive alarms [8, 9]. The Signature-based IDS uses pattern matching with predefined signatures taken from the already detected malware's stored in a database. Thus, creating a low number of false positive alarms but at the same time, it lets new attacks to pass-through unnoticed [10, 11, 12]. Therefore, a system needs to be developed that can increase detection rate for new (a.k.a.zero-day malware's) attacks and reduce false alarms rate in previously defined signatures.

Figure 1 depicts the interaction between the power generation units, distribution centers and other different entities such as industries, smart buildings, households, etc. The smart grid plays a major role in efficiently dissipating the right amount of power to these various entities. The flexibility in the power distribution process is achieved by means of implementing various AI algorithms in the smart grid. The flexibility comes into picture due to the dynamic power requirement from various sectors.

This research uses optimized feature selection technique to detect and classify network intrusions using Signature-based IDS while reducing false alarm rate.Typically, real-time traffic and patterns contain high dimensional space of features. Therefore, feature selection is commonly used to reduce the dimensionality in order to simplify a data set and identify relevant features without sacrificing predictive accuracy. An efficient feature selection can help in cleaning the real-time traffic from noise and irrelevant features [13, 14, 15]. Particle Swarm Optimization (PSO) is a commonly used technique for feature selection [16, 17, 18]. Easy to encode features, support for global searching, requirement of less computational power, fewer parameters and ease of use makes it a common choice of researchers [19, 20, 21]. Therefore, we have used PSO for feature selection in our experiments as well.

Machine-learning algorithms have been commonly used to detect and identify various types of attacks. In this paper, we have implemented several machine-learning algorithms to classify network packets into malicious or normal packets. The novel contribution of this research includes: Modification in the weights of particle swarm optimization

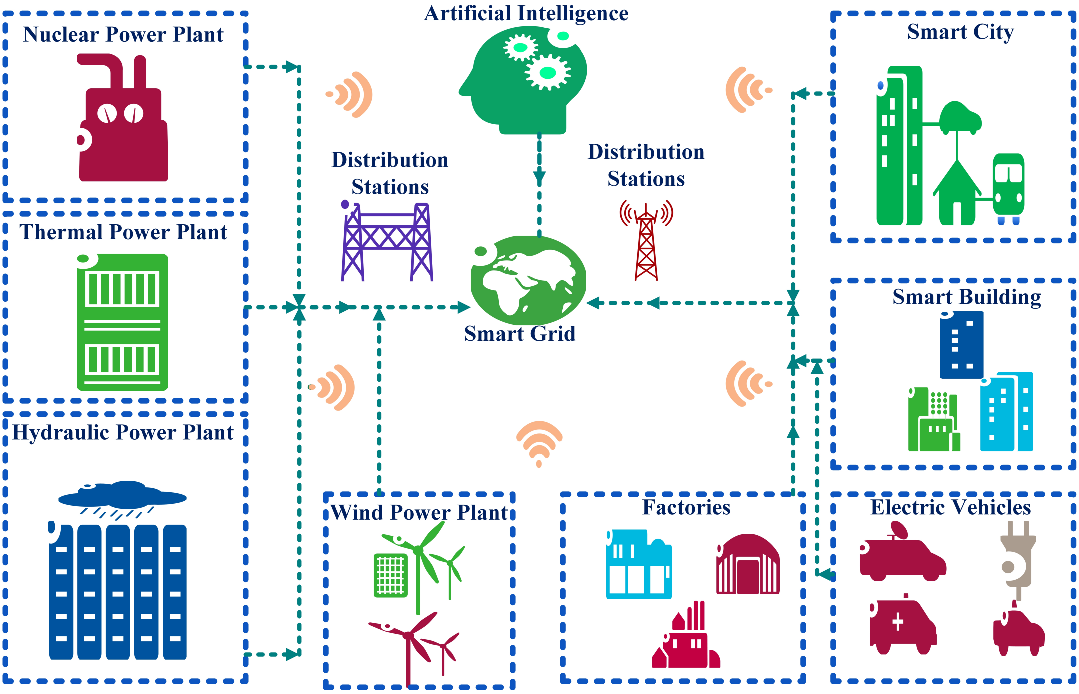

FIGURE 1 Smartgrid System Illustration

algorithms, allowing our proposed weighted particle swarm optimizer to select best features from data sets and those optimal features produces high detection rate, high accuracy and improved false alarm rate. In this research two data sets are used NSLKDD [22] and KDD99 [23]. After selection of data sets some prepossessing techniques are applied on both the data sets. Data sets are normalized using min-max normalization technique in order to scale the data. After data normalization data encoding is performed to convert nominal values to numeric values because machine learning works on numeric data. The proposed system performance is evaluated in terms of accuracy,intrusion detection rate and false alarm rate. The obtained results show that the Random Forest and a Neural Network classifiers have performed better. We have achieved a 0.5% false alarm rate on KDD99 and a 0.08% false alarm rate on the NSLKDD dataset. The detection rate and the testing accuracy on average are 99 % for both datasets.

Paper Organization: Section II evaluates the existing studies and their possible limitations. Section III describes the proposed methodology and techniques adopted, followed by the experiments performed and results tabulated in Section IV. Section V concludes the paper.

2 | RELATED WORK

The demand for electricity is rising day by day and it is estimated that electricity will increase by 30 to 40 percent over the next 20 years. Current power grids are very old; becoming more and more overloaded, unreliable and does not produce enough of electricity. A smart grid has an analytical and well-organized approach to the management of energy supply and usage. The smart grid tracks and regulates the flow of energy in two ways. The consumers

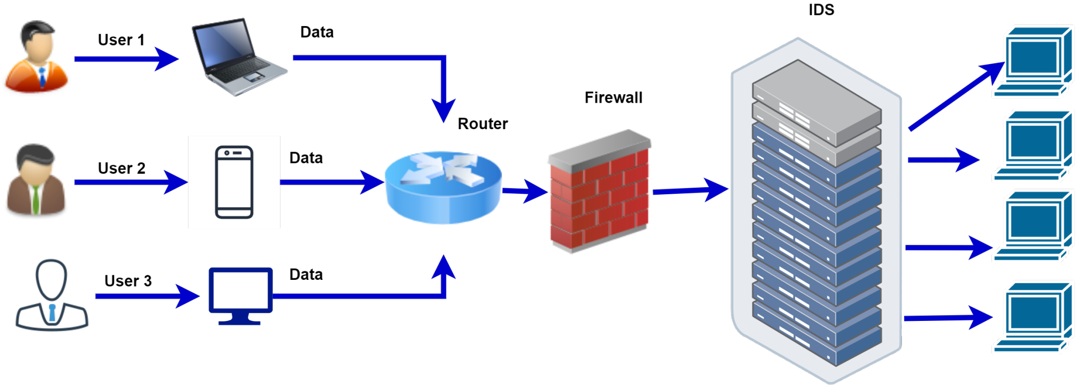

FIGURE 2 Intrusion detection system working

also had the option to use an optimized algorithm to buy the cheapest energy at a particular time, depending on the amount of power used.The smart grid facilitates bidirectional contact between energy suppliers and their clients. The transformation from the current power grid to the smart grid requires new funding, which guarantees the returned great value. The smart grid needs reliable, stable, cost-effective, efficient, environmentally sustainable and healthier facilities.

The smart grid has the below seven key features: allow active customer involvement; manage all production and storage options; create new products, utilities, and markets; offer the best digital economy with power reliability; use energy, optimization, and reliability; ability to self-heal and robust cyber and physical attack actions. The development of smart grids required the integration of diverse technologies and applications. The smart grid has four milestones: customer allowing, advanced delivery operations, advanced transmission operations, advanced asset management. By improving network-wide reliability and dynamic performance, the smart grid increases monitoring and control of the power system co-ordinates. Cyber protection is essential for automatic electric power system operation.

One of the first attempts to achieve a high detection rate and a reduced false alarm rate has been performed on the DARPA 1998 dataset [24].In this paper, authors have used Principal Component Analysis (PCA) to select features and neural networks for classification. Though PCA provides an optimal feature set, it compromises the training efficiency with correct results [25]. Another method for optimal feature selection has been used is Feature Vitality Based Reduction Method (FVBRM) algorithm [19]. The experiment has used 41 features on the NSLKDD dataset using the Naïve Bayes classifier. Some experiments have used multiple techniques for feature selection. Hee-Su et al. [26] have used four feature selection techniques. These techniques are Gain Ratio (GR), Correlation-based Feature Selection (CFS), Information Gain (IG) and Attribute Ratio (AR).

22 Features have been selected from the NSLKDD dataset and for classification, the J48 classifier has been used. Genetic Principal Component (GPA) [27] approach has been used to select optimal features from the KDDCUP99 dataset with SVM classifier for intrusion detection. In order to develop an intelligent IDS using the NSLKDD dataset, Manekar et al. [28] used parameter turning using Particle Swarm Optimization (PSO) with SVM classifier. Another variant of PSO is the intrusion feature selection algorithm (IFSA) based PSO [29, 30]. Which represents velocity and position in intervals compare to a single numeric value. The technique has been used on the KDD99 dataset, while random based PSO has also been used for intrusion detection [31]. PSO can improve the performance of the Multiple Criteria Linear Programming (MCLP) classifier [32]. PSO provides a selection of optimal features for various datasets such as KDDCUP99 [33]. We have investigated various feature selection techniques and performed an analysis of

the available systems that can classify a packet into normal or anomaly classes automatically. We have examined the available literature using the following criteria,as shown in table 1.

TABLE 1 Survey on feature selection and classification techniques

Author	Year	Feature selection	Features	Classifier	Dataset
Heba et.al.[25]	2010	PCA	23	SVM	NSLKDD
Mukherjee et.al.[34]	2012	FVBRM	24	Naïve Bayes	NSLKDD
H.Chae et.al.[26]	2013	AR	22	J48	NSLKDD
		CFS	25		
		IG	23		
		GR	19		
Tesfahun et.al.[35]	2013	IG	22	Random Forest	NSLKDD
Eesa et.al. [27]	2014	RAW	RAW 38	SVM	KDD99
		PCA	PCA 38 PCA 22		
		GPC	GPC 12 GPC 10		
V.Manekar et.al.[28]	2014	PSO	-	SVM(RBF)	NSLKDD
Shrivas et.al.[36]	2014	GR	35	ANN+Bayesian Net	NSLKDD
Patel et.al.[31]	2015	PSO	-	-	NSLKDD
Ahmad et.al.[37]	2015	PCA + PSO	8	MNN	NSLKDD
Eesa et.al.[38]	2015	CFA	5	Decision Tree	KDD99
K.Rai et al [39]	2016	Information Gain	16	DTS	NSLKDD
Bamakan et al.[40]	2016	FMIFS	19,18,4	LSSVM	KDD99 NSLKDD Kyoto2006
Bamakan et al.[41]	2016	TVCPSO	17	SVM	NSLKDD
Thaseen et.al.[42]	2017	Chi	31	SVM	NSLKDD
Syarif et.al.[33]	2017	PSO	25	KNN	KDD99
Pajouh et.al.[43]	2018	-	41	Deep Learning	NSLKDD
Shone et.al.[44]	2018	-	41	RNN	NSLKDD
Naseer et.al.[45]	2018	-	41	LSTM	NSLKDD
Sakr et.al.[46]	2019	BPSO + SPSO + SVM	23	SVM	NSLKDD
Woo et.al.[47]	2019	Correlation Method	40	Neural Network	NSLKDD

From Table 1, we can conclude that though, PCA provides an optimal feature set, but it compromises the training efficiency [48]. The problem with information gain and Gini-index is it give biased results for non-numeric values [49]. Similarly with genetic algorithm and fuzzy logic does not provide surety for optimal solutions [50]. Therefore, more robust solutions are required, which not only give optimal solutions but also have a fast convergence rate, unlike the genetic algorithm, which has a slow convergence rate, also depends upon the population used [51]. That's why we used weighted PSO for feature optimization to make the system more robust. PSO will automatically provide a set of optimal features regardless of the dataset. The above mention feature selection methods either improved detection rate, accuracy, or false alarm rate not all the measures at the same time and on different datasets. These feature selection methods are data-dependent. Therefore, a more optimal way is required, which can solve the above mention problems and perform well regardless of the dataset. For this reason we have proposed, weighted PSO in this research, which achieved promising results compare to other studies.

3 | PROPOSED MODEL

This research proposes an artificial intelligence (AI) base solution for the data-driven security part of the smart grid system by using the optimal features subset and AI models. The objective of this research is to propose a machine learning model which detects network traffic packets quickly and accurately while achieving a low False Alarm Rate (FAR) and high Detection Rate (DR). To achieve this objective optimal feature selection is very important to be used. In this research, the PSO search algorithm is implemented to select the best features from a given subset of features. The datasets used in this research are NSLKDD and KDD99. For both KDD99 and NSLKDD datasets, we perform binary classification, i.e., anomaly or normal, as well as multiclass classification to predict attack categories, such as Denial of Service (DoS), R2L, U2R, Probe and Normal class. After a successful classification of the attacks, we do further classification to handle the exact name of the anomaly. The proposed model consists of six phases. The ***1st*** is data reading, in the data reading phase, we read KDD99 and NSLKDD datasets one by one. The ***2nd*** is data preprocessing, in the preprocessing step, we replace missing values by mean, remove the outliers in data if any, after that data normalization is performed to scale the data. After completing the data normalization, then we performed data encoding to convert non-numeric values into numeric values. The last stage of data preprocessing is the optimal feature selection, which is performed using PSO. The complete working of PSO is discussed in the next section. The ***3rd*** is passing optimal features to machine learning selected models. In the ***4th phase***, we trained different models by passing 70% data and labels to the model. Testing is performed on 30% of the data. ***5th phase*** phase is the experiment phase and ***6th phase*** phase is evaluation.Figure 3 represent the proposed model.

3.1 | Datasets

3.1.1 | KDD99 dataset

KDD99 is one of the most famous datasets used in the field of network security for IDS. KDD99 is a derived version of the 1998 DARPA. It is developed in the MIT research lab and is used by IDS designers as a benchmark to evaluate various methodologies and techniques [52, 53]. KDD99 has 4,900,000 rows and 41 attributes having binary labels and 22 network attacks are listed in the KDD99 dataset. Class labels consist of 4 major attacks like DoS, Probe, U2R, R2L and Normal class.

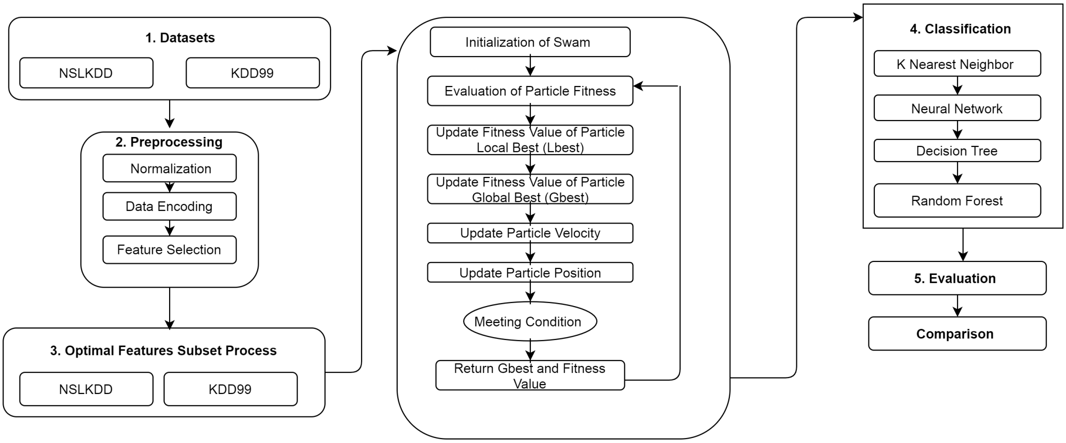

FIGURE 3 Proposed methodology

TABLE 2 KDD99 dataset normal and anomaly packets

Normal packets	97277
Anomaly packets	396731
Total size	494008

Table 2 represents the total number of normal and anomaly packets contain the KDD99 dataset used in this research. 97277 and 396731 packets are used for anomaly and normal class to develop ensemble machine learning classifiers upon which training and testing can be performed. 70% KDD99 dataset is used for training and validation purpose and the rest of the 30% dataset is used for testing and validation, respectively.

3.1.2 | NSLKDD dataset

NSLKDD is an updated copy of the KDD99 dataset. NSLKDD does not have any duplicate Values, which is in the KDD99 dataset. NSLKDD also does not have any inconsistent values. NSLKDD contains 148517 instances and 41 features for training and testing purposes overall.

TABLE 3 NSLKDD dataset normal and anomaly packets

Normal packets	77054
Anomaly packets	71215
Total size	148269

Table 3 represents the total number of normal and anomaly packets contain the NSLKDD dataset used in this research. The total number of an anomaly and normal packets used to train and test machine learning models are 71215 and 77054, respectively. 70% KDD99 dataset is used for training and the rest of the 30% dataset is used for testing and validation, respectively.

TABLE 4 NSLKDD dataset training and testing packets

Training data size	103789
Testing data size	44481

Table 4 represents the total number of an anomaly and normal packets used to train and test machine learning models are 103789 and 44481, respectively. Table 5 represents the number of features in both the datasets.

TABLE 5 Total number of features in KDD99 and NSLKDD datasets

Feature Name	Feature Type	Feature Name	Feature Type
Duration	Number	Protocol type	Non-Numeric
Service	Non-Numeric	Flag	Non-Numeric
Src bytes	Number	Destination bytes	Number
Land	Non-Numeric	Wrong fragt	Number
Urgent	Number	Hot	Number
Num of failed logins	Number	logged in	Non-Numeric
Num access files	Number	Root shell	Number
Su_Attemped	Number	Number root	Number
Number of file creations	Number	Number shells	Number
Number access files	Number	Number outbound commands	Number
Is host login	Non-Numeric	Is guest login	Non-Numeric
Count	Number	Service Count	Number
Serror rate	Number	Service Error rate	Number
Rerror rate	Number	Service error rate	Number
Same service rate	Number	Different service rate	Number
Service different host rate	Number	Dst_host_count	Number
Dst_host_srv_count	Number	Dst_host_same_srv_rate	Number
Dst_host_diff_srv_rate	Number	Dst_host_same_src_port_rate	Number
Dst_host_srv_diff_host_rate	Number	Dst_host_serror_rate	Number
Dst_host_srv_serror_rate	Number	Dst_host_rerror_rate	Number
Dst_host_srv_rerror_rate	Number	Class label type	Non-Numeric

3.2 | Pre-processing

3.2.1 | Normalization:

After selection of dataset, data cleaning operations are performed on datasets to remove noise from dataset and normalize the features. For normalization different techniques are used but in this research min-max normalization approach is used which is better in terms of scaling and solve outliers' issues with z-score normalization Min-max scaling normalizes values in the range of [0, 1]. Equation for min-max normalization is given below.

$$Z_i = \frac{Y_i - \min(Y)}{\max(Y) - \min(Y)} \tag{1}$$

From equation 1, Y=(Y1,Y2,Y3…Yn) are the number of features while Y_i is the feature which we want to normalize and Z_i are normalized features. By doing this now all features have same weights and all features are in one scope.

3.2.2 | Data encoding

Before data encoding, we remove duplicate and inconsistent values from the datasets. Then the nominal attributes are converted to numeric, the reason for that machine learning algorithms back end calculations are done on numeric values not nominal values. So this step is done before passing data to the proposed model.

3.2.3 | Feature selection

Algorithm 1: Steps for PSO algorithm

```
Step1: Randomly set the velocity as well as position of every particle.
Step2: Evaluation of particle fitness.
if fitness value of Pi >Lbesti then
    Lbesti = Pi
else
    if fitness value of Lbesti >Gbesti then
        Gbesti = Lbesti
    else
        Step 3: particle i velocity is updated at this step.
        $\mathrm{D}_{id}^{n+1} = W \times D_{id}^{n} + a_{1\times} r_{1i} \times \left\{L_{id} - P_{iN}^{n}\right\} + C_{2\times} r_{2i} \times \left\{L_{gd} - P_{iN}^{n}\right\}$
        After updating the velocity, position of particle i is updated
        $\mathrm{P}_{id}^{n+1} = P_{id}^{n} + D_{id}^{n+1}$
        Step 4: If threshold for stopping id not achieved then repeat step 2 and step.
        Step 5: At the end, system returns Gbest and its fitness values.
    end
end
```

After feature normalization next important step is the feature optimization. Optimal features not only improve accuracy, but also improve detection rate and false alarm rate. The main focus of feature optimization is to find such feature subsets that can work with different classifiers to produce better results. In this research, we use PSO search method for feature selection. Eberhart and Kennedy [54] in 1995 inspired from fish and birds flock movement behavior and proposed PSO which is generally an optimization algorithm. To solve non-smooth global problems PSO is considered one of the powerful technique [51].Convergence rate of PSO is also very high and it gives optimal solution in less amount of time [55]. Genetic algorithms are also used for optimal feature selection which produce good detection rate but the issue with genetic algorithms is that their convergence rate is very slow and may become worse if subjects of the population are also used [56]. The swarm particles are randomly initialized and then passed to search arena, by changing the value for velocity and for position of particle we can get optimal features subset. The present position and its velocity are expressed in (2) and (3).

$$P_i = \{P_{i1}, P_{i2}, P_{i3}, P_{i4}, P_{i5}, P_{i6} \ldots\ldots .P_{iN}\} \tag{2}$$

Where the dimension of principal search space is represented by N.

$$D_{j=} \left\{D_{j1} D_{j2} D_{j3} D_{j4} D_{j5} D_{j6} \ldots\ldots D_{jN}\right\} \tag{3}$$

Until we get the optimal values algorithm keep updating values for velocity as well as for position. As soon as we get the optimal features, the algorithm stops.

3.3 | Selected optimal features for NSLKDD and KDD99 datasets

TABLE 6 NSLKDD selected optimal attributes

S. No	Feature Name	Data Type
1	Service	Nominal
2	Destination bytes	Numeric
3	Logged-in	Numeric
4	Count	Numeric
5	Srv-diff-host-rate	Numeric
6	Dst-host-count	Numeric
7	Labels	Nominal

TABLE 7 KDD99 selected optimal attributes

S. No	Feature Name	Data Type
1	Service	Numeric
2	Destination bytes	Numeric
3	Logged-in	Numeric
4	Count	Numeric
5	Srv_diff_host_rate	Numeric
6	Dst-host-count	Numeric
7	Dst-host-srv-diff-host-rate	Numeric
8	Labels	Nominal

Table 6 and table 7 represents the optimal features selected form NSLKDD and KDD99 datasets.

3.4 | Classifiers

3.4.1 | K-Nearest Neighbor

K-Nearest Neighbor Classifier (KNN) uses similarity measures to predict new data points. The reason for using the KNN algorithm in this research is that it depends upon the features' similarity. To achieve optimal results, the selection of the right value of K is significant. The value of K is the number of nearest neighbors that are considered in the classification of a vector. In this research, we select K=5, leaf-size=30 and Minkowski metric is used along weights are uniformed. Equations for KNN is given below

$$\text{Eculidean equation } = \sqrt{\sum_{i=1}^{k}(Xi - Yi)^2} \tag{4}$$

$$\text{Manhattan equation} = \sum_{i=1}^{k}|Xi - Yi| \tag{5}$$

$$\text{Minkowski} = \left(\sum_{i=1}^{k}(|X - Yi|)^q\right)^{1/q} \tag{6}$$

3.4.2 | Neural Network (NN)

An NN is a data processing paradigm that is motivated by the biological sensory system. Such as the human brain. The neural network is also widely used in IDS and it is represented in figure 4. Given an input node X_a, the output of the hidden node O_b is given as:

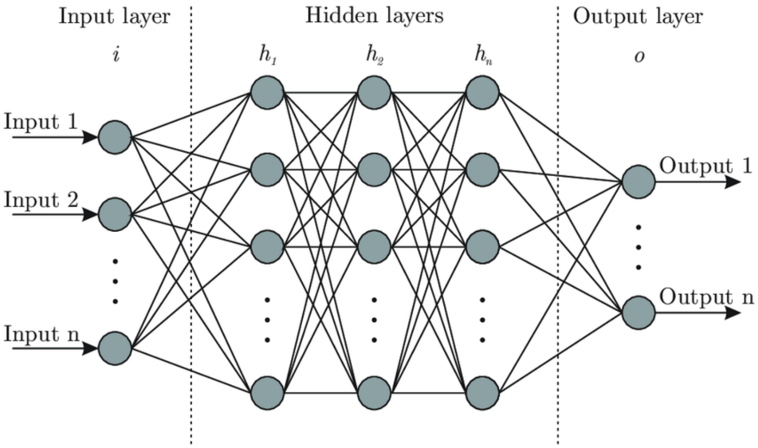

FIGURE 4 Neural Network structure [57]

$$O_b = \phi_1 + \left(\sum_{a=1}^{n} U_{ab} + \theta_b \right) \tag{7}$$

where wa, b represents the weight between the ath input and j_{th} hidden node, and θ_j represents the bias value. Whereas, output will be given

$$\text{output} = \phi_2 + \left(\sum_{b=1}^{n} U_{bk} + \theta_k \right) \tag{8}$$

The mapping of inputs to outputs is an iterative process, where in each iteration weights Ua,b are updated. One of the commonly used algorithm is Back Propagation algorithm which updates the weights using:

$$U_{ba}(t+1) = U_{ba}(t) - \epsilon \frac{\partial Ef}{U_{ba}} \tag{9}$$

The NN is mostly used to solve complex problems and it consists of the input layer, weighted (hidden layers) and output layers. Weights are assigned to each layer in the neural network system. The activation function is also used in the neural network. The NN Model is represented in figure 4. A neural network consists of 60 hidden layers with an activation function of relu, and alpha size is 0.0001. We kept the batch size constant. Max-Iter is 200 and randomness is true.

3.4.3 | Decision Tree

Another algorithm used in recent anomaly-based IDS research is the Decision Tree (DT), this is the same as any tree structure consisting of edges, nodes, leaves etc. A feature and threshold is typically applied to a node and the data is split down the tree, where for example if the data is below a threshold it goes left and above a threshold goes right, until it ends up in a final cluster or class [18]. One DT method is ID3 algorithm that quantifies information by using entropy. Equations for entropy is given below

$$\text{Entropy: } H\,(p_1, p_2, \ldots p_2) = \sum_{i=1}^{s} (p_i \log{(1/p_i)} \tag{10}$$

Where (p1,p2,...ps) represents the probabilties of the class labels.

$$\text{Gain}(D, S) = H(D) \sum_{i=1}^{s} p\,(D_i)\,H\,(D_i) \tag{11}$$

Another decision tree method is called the C4.5. Decision tree [58, 59] has the ability to process large amounts of data efficiently is used to sort data into groups so that a Support Vector Machine (SVM) can classify the smaller subsets of information. In [60] author proposed a similar method however an SVM is placed on each edge in the DT. We performed splitting using gini-index, max-depth=none, min-samples-split=2, min-samples-leaf=1, class-weights=none, random-state=none, min-impurity-decrease=0.0 and min-impurity-split=none.

3.4.4 | Random Forest

Random Forest classifier plays a significant part in IDS. It is a combination of multiple decision trees and random forest combine all the decision trees to get prediction sharpened and get more accurate results. The best thing about the random forest is that it can be used for both regression and classification. The random forest also tells us about the importance of the features that will help in deciding which features should be kept and which ones should be dropped from the dataset.

3.5 | Evaluation metrics

Various performance metrics are used to evaluate the proposed solution, including precision, recall, F1-Measure [61], False Alarm Rate (FAR), Detection Rate (DR) and Accuracy. Above mention performance metrics base on True Positive (TP), False Positive (FP), False Negative (FN) and True Negative (TN).

False Alarm Rate (FAR) is a combination of total instances that are normal but classify as attack class and truly classify attack class.

$$FAR = \frac{FP}{FP + TN} \tag{12}$$

Accuracy [62] is used to measure how many instances are correctly classified as normal and attacks classes. Accuracy is achieved by summing correctly classify instances with dividing the total instances represented in equation 13.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{13}$$

Detection Rate (DR) provides information about the attacks detected correctly divided by the total number of attacks in the dataset.

$$DR = \frac{TP}{TP + FN} \tag{14}$$

Precision's objective is to evaluate the True Positive (TP) entities in relation to False Positive (FP) entities.

$$Precision = \frac{TP}{TP + FP} \tag{15}$$

The purpose of recall is to evaluate True Positive (TP) entities in relation to (FN) False Negative entities that are not at all categorized. The mathematical form of recall is mentioned in equation (16).

$$Recall = \frac{TP}{TP + FN} \tag{16}$$

Sometimes performance assessment may not be good with accuracy and recall, For instance, if one mining algorithm has low recall but high precision that another algorithm is needed. Then there is the question of which algorithm is better. This problem is solved by using F1-score that gives an average recall and precision. F1-score can be calculated as shown in equation (17).

$$F1 - score = \frac{2{*}\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{17}$$

4 | EXPERIMENT RESULTS

In this section experiment results of KDD99 and NSLKDD are mentioned. All these experiments are performed on google colab. System specification core I3 system with 8 GB RAM and 2.7 GHz processor is used.

TABLE 8 Classification report for KDD99

Model Name	Class	Precision %	Recall %	F1-score %
PSO + KNN	Normal	98.8	97.6	98.2
	Attack	99.4	99.7	99.6
PSO + Neural Network	Normal	95.4	99.6	97.5
	Attack	99.9	98.8	99.4
PSO + Decision Tree	Normal	98.5	99.2	98.8
	Attack	99.8	99.6	99.7
PSO + Random Forest	Normal	98.5	99.3	98.9
	Attack	99.8	99.6	99.7

From Table 8, we can conclude that precision, recall and f1-score for KNN, normal class is 98.89%, 97.60%, 98.20%, respectively. Similarly, for an anomaly class, precision is 99.40%, the recall is 99.70% and the f1-score is 99.60%, respectively. Random forest precision, recall and f1-score for the normal class will give us 98.50%,99.30%,98.90%, respectively. Precision, recall and f1-score for attack class are 99.80%, 99.60%,99.70%. For decision tree and neural network, precision scores for the normal class are 98.50%, 95.40%, respectively. Similarly, recall and f1-scores are 99.30% and 98.40% on average for a normal class. Precision recall and f1-scores on average for an attack class using decision tree and neural network are 97%, 99.60% 99.50% respectively depicted in the figure 5.

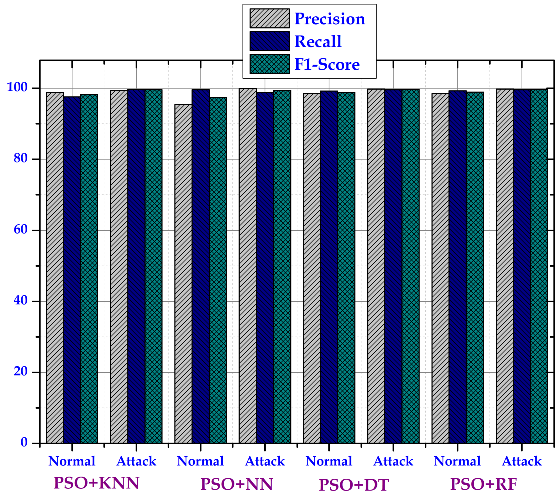

FIGURE 5 Classification report for KDD99 datasets

TABLE 9 FAR, DR and Accuracy comparison report

Model Name	KDD99 (FAR %)	NSLKDD (FAR %)
PSO + KNN	2.40	0.17
PSO + Neural Network	0.50	3.13
PSO + Decision Tree	0.80	0.14
PSO + Random Forest	0.60	0.08

Table 9 and figure 6 depicts that the KNN classifier with KDD99 dataset achieved 2.4% FAR which is high compare to other classifiers, decision tree and random forest achieved 0.8% and 0.6% FAR respectively. For KDD99 neural network outperformed other classifiers in terms of FAR and it achieved 0.5% FAR. The reason for this is neural network performs well on large dataset and KDD99 dataset has more data compare to NSLKDD dataset. Similarly random forest achieved promising results for FAR using NSLKDD dataset. FAR for random forest is 0.08%, since random forest is ensemble classifier and it is the combination of multiple decision tree that's why it achieved promising results compare to other classifiers like decision tree, KNN and NN.

From table 10 and table 11 we can conclude that using the KNN classier with KDD99 dataset, 118779 packets are identified as an attack, while only 337 packets are misclassified out of 119116 packets. For normal class out of 29090 packets, 28390 packets are detected correctly and 700 packets are identified incorrectly with the accuracy of 97.60% for normal class and 99.70% for attack class, respectively. The detection rate for the knn classifiers is 99.70%. True positive for random forest and decisions tree are 118672 and 118680, respectively. The true negative for the random forest is 28902. Similarly, for the decision tree the true negative is 28850. False positive and false negative scores for the random forest is 188 and 444, respectively. For the decision tree overall, 676 packets are misclassified. The detection rate for both the random forest and the decision tree is 99.60%, respectively. The neural network also

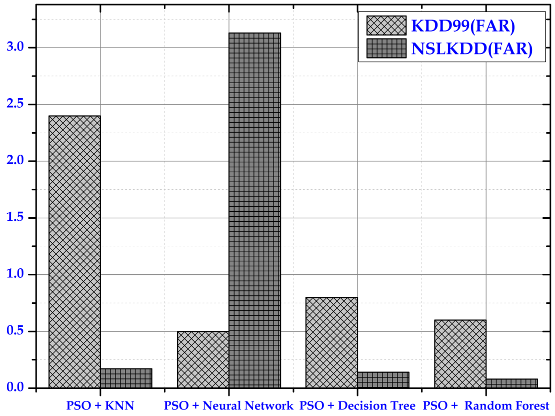

FIGURE 6 FAR for KDD99 and NSLKDD datasets

achieved promising results for true positive and for true negative with the detection rate of 99.20%.118161 packets are correctly detected as an attack with an accuracy of 99.20%, while 28927 packets are correctly identified normal packets with an accuracy of 99.40%. 95 packets are misclassified for attack class and 163 packets for the normal class using a neural network.Figure 7 represents accuracy and detection rate for both datasets.

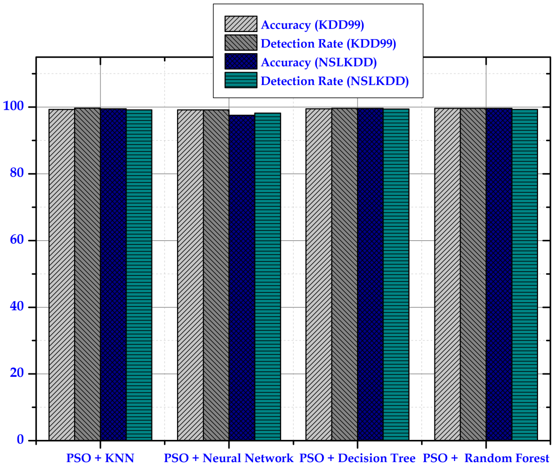

FIGURE 7 Accuracy and DR for both datasets.

TABLE 10 Confusion matrix for KDD99

Model Name	TP	FN	FP	TN
PSO+ KNN	118779	337	700	28390
PSO+ Neural Network	118161	95	163	28927
PSO+ Decision Tree	118680	436	240	28850
PSO+ Random Forest	118672	444	188	28902

TABLE 11 Accuracy and DR for both datasets.

Model Name	KDD99		NSLKDD	
	Accuracy %	**DR %**	**Accuracy %**	**DR %**
PSO+KNN	99.3	99.7	99.51	99.17
PSO+NN	99.2	99.2	97.54	98.18
PSO+DT	99.5	99.6	99.64	99.41
PSO+RF	99.6	99.6	99.65	99.3

TABLE 12 Confusion matrix for NSLKDD

Model Name	TP	FN	FP	TN
PSO+ KNN	21255	176	41	23083
PSO+ Neural Network	21041	390	703	22421
PSO+ Decision Tree	21306	125	34	23090
PSO+ Random Forest	21295	136	20	23104

Table 11 and table 12 represents that the random forest with PSO achieved 99.65% accuracy and 99.30% detection rate, respectively. Precision, recall and f1-scores are 99.40%, 99.90%, 99.70% respectively for a normal class. Similarly, for an anomaly class, we achieved 99.90% precision, 99.40% recall and 99.60% f1-score, respectively. KNN model gained 99.51% accuracy overall, for normal class accuracy is 99.8%, while for an attack class, accuracy is 99.20%. Decision tree detected 21307 packets correctly as anomaly out of 21431 with the accuracy of 99.40% and out of 23124 normal packets, 23093 packets correctly identified as normal traffic with the accuracy of 99.90%. For an attack class decision tree achieved 99.80% precision, recall is 99.40% and 99.70% f1-score, similarly for normal class precision is 99.50% while recall is 99.90% and f1-score is 99.70%. Using a multilayer perceptron, we achieved 99.50% accuracy for normal class and 97.90% accuracy for anomaly class. 98.5% overall accuracy is achieved in [42]. Similarly, in [61] they got 97.87% overall accuracy. We gained a 98.18% detection rate while the false alarm rate is around 3.13% using a multilayer perceptron. MLP results are a little low compare to knn, decision tree and random forest, the reason for this is a neural network performs well when class is balance and when we have a large amount of data for both training and testing. For a normal class preicion, recall and f1-score is 95.10%, 99.90% and 97.40%

respectively using multilayer perceptron classifier and NSLKDD dataset. Similarly, for an anomaly class, precision, recall and f1-score is 99.90%, 94.50%,97.10%, respectively, depicated in table 13 and figure 8.

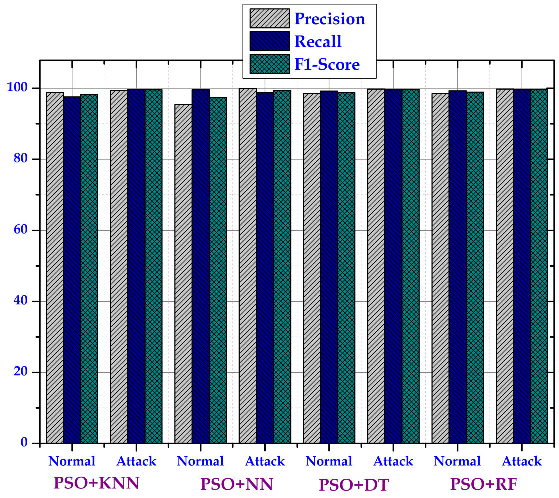

FIGURE 8 Classification report for NSLKDD dataset.

TABLE 13 Classification report for NSLKDD

Model Name	Class	Precision %	Recall %	F1-score %
PSO + KNN	Normal	99.2	99.8	99.5
	Attack	99.8	99.2	99.5
PSO + Neural Network	Normal	95.1	99.9	97.4
	Attack	99.9	94.5	97.1
PSO + Decision Tree	Normal	99.5	99.9	99.7
	Attack	99.8	99.4	99.6
PSO + Random Forest	Normal	99.4	99.9	99.7
	Attack	99.9	99.4	99.6

4.1 | KDD99 Multi Class Classification Experimental Results

Table 14 and figure 9 depict that normal class achieved 98.30% precision, 96.10% recall and 97.10% F1-Measure, respectively. TP and FP rate is 96.10% and 0.4% respectively. Smurf and Warezclient achieved a 100% detection rate, respectively. Similarly, for Warezclient and Smurf attack has 0% and 0.3% FP rate, respectively. Recall for both Warezclient and Smurf attacks is 100%, respectively, while f1-score is above 99% on average for both the attacks, respectively. Precision for Warezclient is 99.30% and Smurf precision is 98.9%, respectively, for Portsweep DR and recall is 89.20%, respectively. FP rate for Portsweep is high compare to other attacks using a decision tree, which is around 1.8%. Precision and F1-Measure scores are 77.20% and 82.80% respectively for Portsweep. On average, precision, recall, F1-Measure and TP rate scores for Ipsweep are 98.50% and the FP rate is 0.2%, respectively. Saran, Nmap, Back, Teardrop and Neptune also performed well and achieved, on average, 93% precision, recall and F1-Measure, respectively.

TABLE 14 Classification report for Decision Tree

Class	TP Rate %	FP Rate %	Precision %	Recall %	F1-score %
saran	84.7	0.3	97.3	84.7	90.6
portsweep	89.2	1.8	77.3	89.2	82.8
ipsweep	99.1	0.2	96.7	99.1	97.9
nmap	41.2	0	96.6	41.2	57.7
back	97.9	0.1	98.9	97.9	98.4
teardrop	86.1	1.5	76.6	86.1	81.1
warezclient	100	0	99.3	100	99.7
neptune	95.6	1.7	92.8	95.6	94.2
smurf	100	0.3	98.9	100	99.4
normal	96.1	0.4	98.3	96.1	97.1

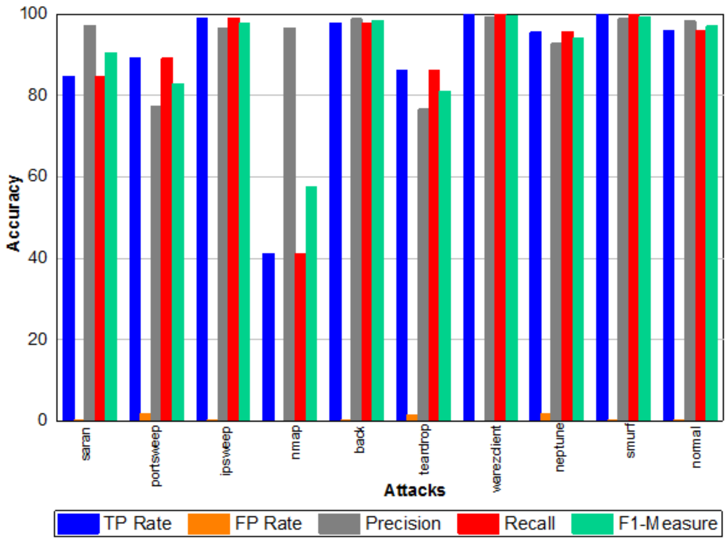

FIGURE 9 Classification report for Decision Tree

TABLE 15 Classification report for Random Forest

Class	TP Rate %	FP Rate %	Precision %	Recall %	F1-score %
saran	85.1	0.5	95.3	85.1	89.9
portsweep	89.5	1.9	76.5	89.5	82.5
ipsweep	99.1	0	99.4	99.1	99.2
nmap	42.6	0	96.7	42.6	59.2
back	97.9	0.1	99.2	97.9	98.5
teardrop	81.1	1.6	73.9	81.1	77.3
warezclient	100	0	99.3	100	99.7
neptune	96.1	1.7	92.6	96.1	94.3
smurf	100	0	99.9	100	99.9
normal	96.3	0.4	98.2	96.3	97.2

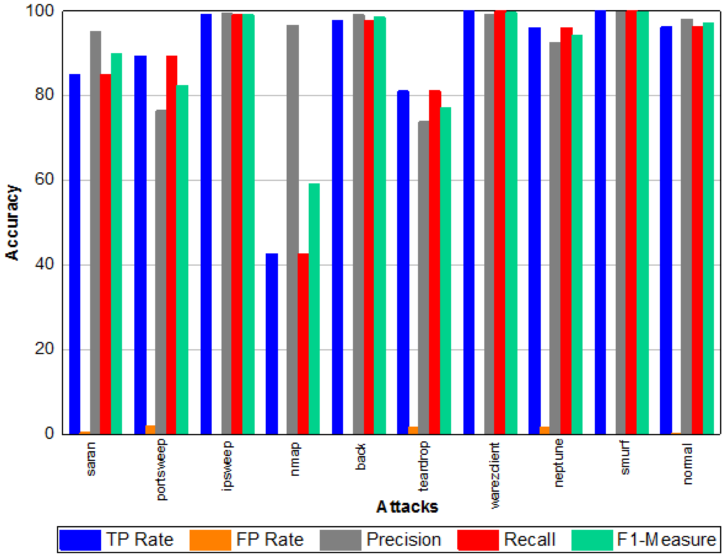

FIGURE 10 Classification report for Random Forest

From table 15 and figure 10, we can conclude that the FR rate for Ipsweep, Nmap, Warezclient and Smurf is 0%, respectively, which is promising. Similarly, the DR rate for those attacks is 99.10, 42.60%, 100%, respectively. Saran, Portsweep, Back, Teardrop and Neptune achieved 0.5%, 1.9%, 0.1%, 1.6% , 1.7% FR rate respectively.The DR rate for those attack is 85.10%, 89.50%, 97.90%, 81.10% ,96.10% respectively.Precision, recall and F1-Measure for all attacks on average are 92.50%, 87.93% ,88.88%, respectively. For normal class precision, recall and F1-Measure is 98.20%, 96.30%, 97.20%, respectively.TP and FP for normal class is 96.30% and 0.4%, respectively.

TABLE 16 Classification report for K Nearest Neighbour

Class	TP Rate %	FP Rate %	Precision %	Recall %	F1-score %
saran	84.9	0.5	94.6	84.9	89.5
portsweep	90.5	1.9	76	90.5	82.6
ipsweep	99.1	0	100	99.1	99.5
nmap	42.6	0	96.7	42.6	59.2
back	97.6	0.2	98.1	97.6	97.9
teardrop	83	1.7	73.9	83	78.2
warezclient	100	0	100	100	100
neptune	95.4	1.6	93.3	95.4	94.3
smurf	100	0	100	100	100
normal	95.8	0.5	97.7	95.8	96.8

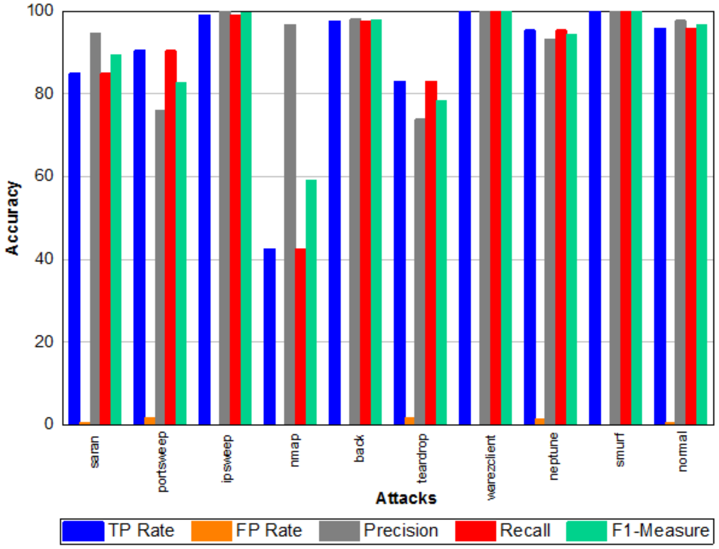

FIGURE 11 Classification report for K Nearest Neighbour

For the table 16, we can conclude that Saran attack, TP, FP, precision, recall and f1-score is 84.9%, 0.5%, 94.6%, 84.9%,89.5% respectively. Portsweep has 90.50%, 1.9%, 76%, 90.50%, and 82.60% TP, FP, precision, recall, f1-score respectively. TP rate for Ipswep, Back and Neptune attacks is 99.10%, 97.60%, 95.40%, respectively. Similarly, FP rate for those attacks is 0%, 0.22,1.6%, respectively. The precision for Ipsweep is 100%. Recall and F1-Measure for Ipsweep is 99.10%, 99.50%, respectively. Precision for Back and Neptune is 98.10%. 93.30%, respectively. For back attack recall and f1-score is 97.60%,97.90%, respectively. Similarly, for Neptune, it is 95.40% and 94.30%, respectively, for the Nmap TP rate and the recall score is 42.60%, respectively. FR rate is 0%. Precision and recall scores are 96.70% and 59.20%, respectively. Warezclient and Smurf attack achieved promising results using the KNN classifier. Precision, recall, f1-score and TP rate are 100% respectively for both attacks. The normal class achieved, on average, 95% TP, precision, recall and f1-score, respectively, depicted in figure 11.

TABLE 17 Classification report for Neural Network

Class	TP Rate %	FP Rate %	Precision %	Recall %	F1-score %
saran	84.3	0.5	94.8	84.3	89.2
portsweep	91.8	5.9	51.1	91.8	65.7
ipsweep	99.1	0	99.4	99.1	99.2
nmap	42.6	0.1	90.6	42.6	58
back	98.7	1.1	88.6	98.7	93.4
teardrop	62.5	1	78.3	62.5	69.5
warezclient	100	0.1	99	100	99.5
neptune	96.7	1.6	93.3	96.7	95
smurf	100	0.1	99.7	100	99.8
normal	79.2	0.3	98.4	79.2	87.8

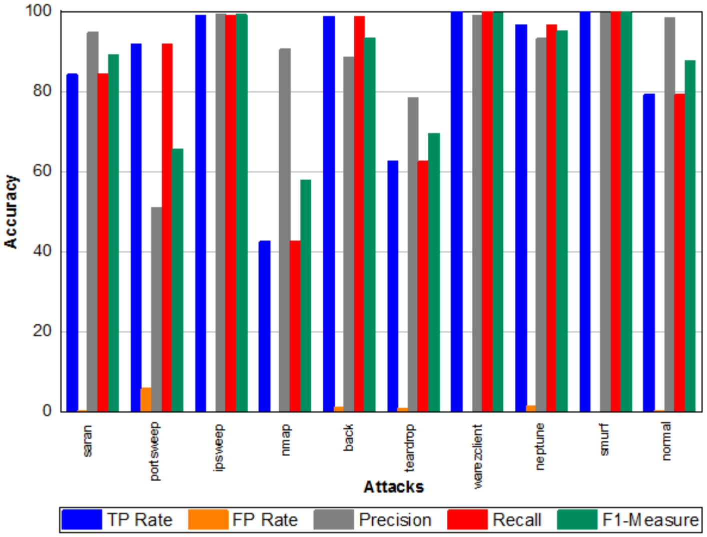

FIGURE 12 Classification report for Neural Network

From table 17 and figure 12, we conclude that the TR rate for attacks and the normal class is 95.36% on average.

Similarly, the average FP rate is 1.06% for all the classes in NSLKDD dataset. Average precision, recall and F1-Measure scores are 89.32%, 85.49%, 85.33% respectively for all the attacks and normal class using decision tree algorithm.

4.2 | NSLKDD Multi Class Classification Experimental Results

TABLE 18 Classification report for Decision Report

Class	TP Rate %	FP Rate %	Precision %	Recall %	F1-score %
warezclient	98.2	0.2	96	98.2	97.1
ipsweep	90.8	0.3	98.4	90.8	94.4
portsweep	99.2	0.1	99.1	99.2	99.1
teardrop	100	0	100	100	100
nmap	97.6	1.3	82.9	97.6	89.7
smurf	100	0.3	97.4	100	98.7
back	98.9	0	99.6	98.9	99.2
satan	98.8	0.2	98.7	98.8	98.8
neptune	99.3	0	99.8	99.3	99.6
normal	96.9	0.3	98.2	96.9	97.5

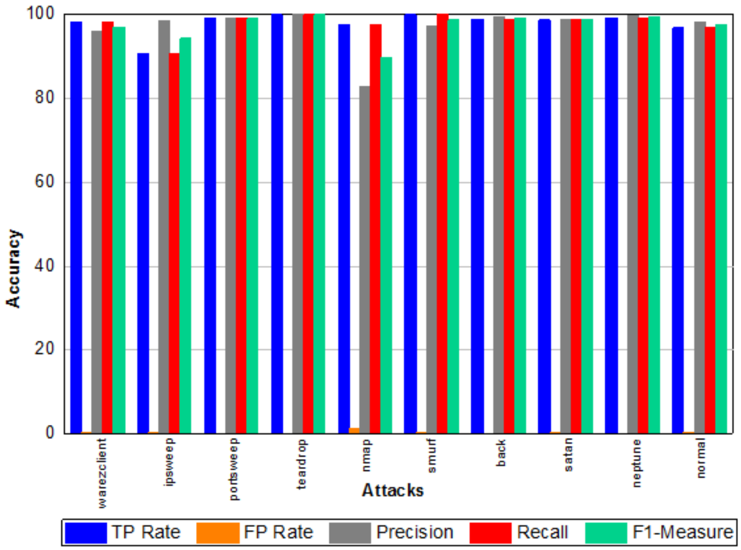

FIGURE 13 Classification report for Decision Tree

From table 18 and figure 13, we can conclude that the TP rate for attacks and the normal class is 95.36% on average. Similarly, the average FP rate is 1.06% for all the classes in NSLKDD dataset. Average precision, recall and f1-measure scores are 89.32%, 85.49%, 85.33% respectively for all the attacks and normal class using decision tree algorithm.

TABLE 19 Classification report for Random Forest

Class	TP Rate %	FP Rate %	Precision %	Recall %	F1-score %
warezclient	100	0.1	98.6	100	99.3
ipsweep	90.8	0.1	99.3	90.8	94.9
portsweep	98.7	0.1	98.9	98.7	98.8
teardrop	100	0	100	100	100
nmap	96.7	1.3	83	96.7	89.4
smurf	100	0.3	97.9	100	99
back	100	0	100	100	100
satan	98.8	0.3	98.3	98.8	98.5
neptune	99.9	0.1	99.2	99.9	99.6
normal	98.4	0.2	99.1	98.4	98.7

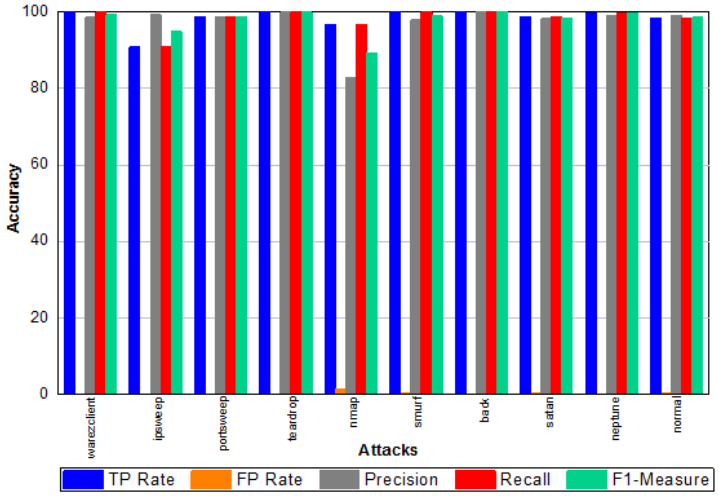

FIGURE 14 Classification report for Random Forest

Table 19 depicts that Warezclient, Teardrop, Smurf and Back attack have a 100% TP rate and 100% recall, respectively. Teardrop and Back attack has a 0% FP rate, respectively. Warezlient, Ipsweep, Portsweep and Neptune have a 0.1% FP rate, respectively. Smurf and Satan have a 0.3% FP rate, respectively. Satan has 0.3% and normal has 0.2% FR rates, respectively. Warezclient, Portsweep and Satan have 98% precision, respectively. Ipsweerp, Neptune and normal have 99% precision, respectively. Portsweep, Neptune and normal class hs 98% recall, respectively. Similarly, Ipsweep, Nmap and Neptune have 90.8%,96.7% and 99.9% recall, respectively. f1-measure for Warezclient, Smurf and Neptune is 99%, respectively. Portsweep, Satan and Normal have 98% f1-measure, respectively. Teardrop and Back have 100% f1-measure, respectively. Nmap has 89.4% f1-measure using a random forest classifier and NSLKDD dataset. The visualization of these attacks is depicted in figure 14.

TABLE 20 Classification report for K Nearest Neighbour

Class	TP Rate %	FP Rate %	Precision %	Recall %	F1-score %
warezclient	99.5	0.2	95.2	99.5	97.3
ipsweep	90.8	0.1	99.3	90.8	94.9
portsweep	97.9	0.1	99.1	97.9	98.5
teardrop	100	0	100	100	100
nmap	97	1.2	83.3	97	89.6
smurf	100	0.3	97.7	100	98.8
back	100	0	99.2	100	99.6
satan	98.7	0.4	97.7	98.7	98.2
neptune	99.3	0.2	98.8	99.3	99.1
normal	96.7	0.3	98.2	96.7	97.4

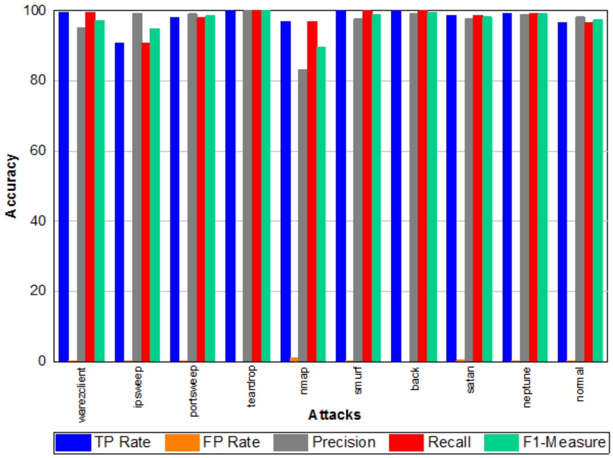

FIGURE 15 Classification report for K Nearest Neighbour

From table 20 and figure 15, we conclude that the TP rate for attacks and the normal class is 97.99% on average. Similarly, the average FP rate is 0.28% for all the classes in NSLKDD dataset. Average precision, recall and f1-measure scores are 97.97%, 97.99% and 97.34 respectively for all the attacks and normal class using KNN algorithm.

TABLE 21 Classification report for Neural Network

Class	TP Rate %	FP Rate %	Precision %	Recall %	F1-score %
warezclient	99.5	0.5	88.6	99.5	93.7
ipsweep	90.5	0.2	99.1	90.5	94.6
portsweep	98.8	0.2	98.8	98.8	98.8
teardrop	100	0	100	100	100
nmap	91.1	1.3	82.2	91.1	86.4
smurf	100	0.3	97.4	100	98.7
back	98.5	7.2	37.7	98.5	54.6
satan	98.4	0.7	95.9	98.4	97.1
neptune	99.4	0.1	99.6	99.4	99.5
normal	46.7	0.4	94.9	46.7	62.6

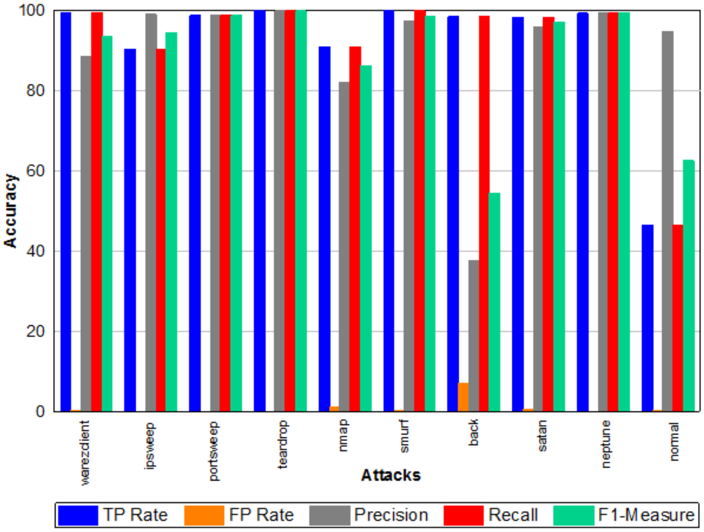

FIGURE 16 Classification report for Neural Network

From table 21 and figure 16, we conclude that the TP rate for attacks and the normal class is 92.29% on average. Similarly, the average FP rate is 1.6% for all the classes in NSLKDD dataset. Average precision, recall and F1-Measure scores becomes 89.44%, 92.25% and 88.6% respectively for all the attacks and normal class using decision tree algorithm.

TABLE 22 Comparison of proposed model with other models (KDD99)

Model	Accuracy %	FAR %	DR %
PSO+MCLP [32]	99.13	1.94	-
TVCPSO [41]	-	0.80	97
SVM-ELM [63]	95.75	1.87	95.17
PSO [64]	88.5	-	-
DNN [65]	75.5	0.85	76
PSO-ANN [66]	92.5	-	-
ANN(FNN-LSO)	94.02	2.23	89.83
Proposed Model (PSO+NN)	99.20	0.5	99.70

TABLE 23 Comparison of proposed model with other models (NSLKDD)

Model	Accuracy %	FAR %	DR %
RF [43]	93.77	-	-
SVM-ELM [44]	95.75	1.87	95.17
DNEDRON [45]	97.55	1.08	95.97
RNN-IDS [46]	99.81	5.09	96.92
HIERARCHICAL SOM [47]	-	2.19	93.46
ADABOOST [48]	-	3.14	91.20
LSTM [49]	93.82	0.09	77.12
GA [50]	88.77	-	-
Proposed Model	99.65	0.08	99.3

5 | CONCLUSION AND FUTURE WORK

This paper proposes a feature selection base IDS system for smart grid systems. For this purpose, we have used weighted PSO to improve the false alarm rate in the IDS. Optimal features are selected from KDD99 and NSLKDD datasets. After optimal features selection, these features are passed to machine learning models. We have applied various machine learning algorithms on NSLKDD and KDD99 datasets during the experiments. After the collection of datasets, we have transformed them into a binary classification: attack class and normal class as well as we used multiple attacks. 9 attacks are used for the KDD99 dataset. In comparison, 21 attacks are used for the NSLKDD dataset. Initially, we have performed preprocessing on the datasets and non-numeric values are replaced with numeric encoding. Next, the data is normalized using min-max normalization. After that, we have performed feature selection using particle swarm optimization and selected the best features. After feature selection, we have applied different machine learning algorithms on both the datasets. Random Forest and Neural Network have outperformed all other methods in terms of accuracy, training time and false alarm rate. We have also compared our proposed methodology with other recent work as shown in Table 22 and Table 23. Experimental results prove that our method performs better in terms of detection rate, false alarm rate and accuracy for both KDD99 and NSLKDD datasets. In future, we intend to repeat this experiment with multiple classes with feature selection methods using deep learning algorithms.

references

[1] Otoum Y, Liu D, Nayak A. DL-IDS: a deep learning–based intrusion detection framework for securing IoT. Transactions on Emerging Telecommunications Technologies 2019;p. e3803.

[2] Yousaf A, Loan A, Babiceanu RF, Yousaf O. Physical-layer intrusion detection system for smart jamming attacks. Transactions on Emerging Telecommunications Technologies 2017;28(11):e3189.

[3] Irshad O, Khan MUG, Iqbal R, Basheer S, Bashir AK. Performance optimization of IoT based biological systems using deep learning. Computer Communications 2020;.

[4] Vora J, Kaneriya S, Tanwar S, Tyagi S, Kumar N, Obaidat M. TILAA: Tactile Internet-based ambient assistant living in fog environment. Future Generation Computer Systems 2019;98:635–649.

[5] Uppal HAM, Javed M, Arshad M. An overview of intrusion detection system (IDS) along with its commonly used techniques and classifications. International Journal of Computer Science and Telecommunications 2014;5(2):20–24.

[6] Bhattacharya S, Kaluri R, Singh S, Alazab M, Tariq U, et al. A Novel PCA-Firefly based XGBoost classification model for Intrusion Detection in Networks using GPU. Electronics 2020;9(2):219.

[7] Alazab M, Khan S, Siva Rama Krishnan S, Pham Q, Praveen Kumar Reddy M, Gadekallu TR. A Multidirectional LSTM Model for Predicting the Stability of a Smart Grid. IEEE Access 2020;.

[8] Almseidin M, Alzubi M, Kovacs S, Alkasassbeh M. Evaluation of machine learning algorithms for intrusion detection system. In: 2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY) IEEE; 2017. p. 000277–000282.

[9] Iwendi C, Maddikunta PKR, Gadekallu TR, Lakshmanna K, Bashir AK, Piran MJ. A metaheuristic optimization approach for energy efficiency in the IoT networks. Software: Practice and Experience 2020;.

[10] Jyothsna V, Prasad VR, Prasad KM. A review of anomaly based intrusion detection systems. International Journal of Computer Applications 2011;28(7):26–35.

[11] Kaur H, Kumar N, Batra S. ClaMPP: a cloud-based multi-party privacy preserving classification scheme for distributed applications. The Journal of Supercomputing 2019;75(6):3046–3075.

[12] Aujla GS, Kumar N, Singh M, Zomaya AY. Energy trading with dynamic pricing for electric vehicles in a smart city environment. Journal of Parallel and Distributed Computing 2019;127:169–183.

[13] Aghdam MH, Ghasem-Aghaee N, Basiri ME. Application of ant colony optimization for feature selection in text categorization. In: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence) IEEE; 2008. p. 2867–2873.

[14] Aghdam MH, Tanha J, Naghsh-Nilchi AR, Basiri ME. Combination of ant colony optimization and Bayesian classification for feature selection in a bioinformatics dataset. Journal of Computer Science & Systems Biology 2009;2(3):186–199.

[15] Nguyen VG, Brunstrom A, Grinnemo KJ, Taheri J, Liyanage M, Ahmad I, et al. 5G mobile networks: Requirements, enabling technologies, and research activities. A Comprehensive Guide to 5G Security 2018;p. 31–57.

[16] Qasim OS, Algamal ZY. Feature selection using particle swarm optimization-based logistic regression model. Chemometrics and Intelligent Laboratory Systems 2018;182:41–46.

[17] Ma T, Xu C, Zhou Z, Kuang X, Zhong L. SE-PSO: Resource Scheduling Strategy for Multimedia Cloud Platform Based on Security Enhanced Virtual Migration. In: 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC) IEEE; 2019. p. 650–655.

[18] Iwendi C, Khan S, Anajemba JH, Mittal M, Alenezi M, Alazab M. The Use of Ensemble Models for Multiple Class and Binary Class Classification for Improving Intrusion Detection Systems. Sensors 2020;20(9):2559.

[19] Xue B, Zhang M, Browne WN. Particle swarm optimization for feature selection in classification: A multi-objective approach. IEEE transactions on cybernetics 2012;43(6):1656–1671.

[20] 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.

[21] Reddy T, RM SP, Parimala M, Chowdhary CL, Hakak S, Khan WZ, et al. A deep neural networks based model for uninterrupted marine environment monitoring. Computer Communications 2020;.

[22] Revathi S, Malathi A. A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. International Journal of Engineering Research & Technology (IJERT) 2013;2(12):1848–1853.

[23] Tavallaee M, Bagheri E, Lu W, Ghorbani AA. A detailed analysis of the KDD CUP 99 data set. In: 2009 IEEE symposium on computational intelligence for security and defense applications IEEE; 2009. p. 1–6.

[24] Liu G, Yi Z, Yang S. A hierarchical intrusion detection model based on the PCA neural networks. Neurocomputing 2007;70(7-9):1561–1568.

[25] Heba FE, Darwish A, Hassanien AE, Abraham A. Principle components analysis and support vector machine based intrusion detection system. In: 2010 10th international conference on intelligent systems design and applications IEEE; 2010. p. 363–367.

[26] Chae Hs, Jo Bo, Choi SH, Park Tk. Feature selection for intrusion detection using NSL-KDD. Recent advances in computer science 2013;p. 184–187.

[27] Ahmad I, Hussain M, Alghamdi A, Alelaiwi A. Enhancing SVM performance in intrusion detection using optimal feature subset selection based on genetic principal components. Neural computing and applications 2014;24(7-8):1671–1682.

[28] Manekar V, Waghmare K. Intrusion detection system using support vector machine (SVM) and particle swarm optimization (PSO). International Journal of Advanced Computer Research 2014;4(3):808.

[29] Tong L, Wu Q. Intrusion Feature Selection Algorithm Based on Particle Swarm Optimization. International Journal of Computer Science and Network Security (IJCSNS) 2014;14(12):40.

[30] Zhang T, Kuang X, Zhou Z, Gao H, Xu C. An Intelligent Route Mutation Mechanism against Mixed Attack Based on Security Awareness. In: 2019 IEEE Global Communications Conference (GLOBECOM) IEEE; 2019. p. 1–6.

[31] Patel R, Bakhshi D, Arjariya T. Random particle swarm optimization (RPSO) based intrusion detection system. International Journal of Advanced Technology and Engineering Exploration 2015;2(5):60.

[32] Bamakan SMH, Amiri B, Mirzabagheri M, Shi Y. A new intrusion detection approach using PSO based multiple criteria linear programming. Procedia Computer Science 2015;55:231–237.

[33] Syarif AR, Gata W. Intrusion detection system using hybrid binary PSO and K-nearest neighborhood algorithm. In: 2017 11th International Conference on Information & Communication Technology and System (ICTS) IEEE; 2017. p. 181–186.

[34] Mukherjee S, Sharma N. Intrusion detection using naive Bayes classifier with feature reduction. Procedia Technology 2012;4:119–128.

[35] Tesfahun A, Bhaskari DL. Intrusion detection using random forests classifier with SMOTE and feature reduction. In: 2013 International Conference on Cloud & Ubiquitous Computing & Emerging Technologies IEEE; 2013. p. 127–132.

[36] Shrivas AK, Dewangan AK. An ensemble model for classification of attacks with feature selection based on KDD99 and NSL-KDD data set. International Journal of Computer Applications 2014;99(15):8–13.

[37] Ahmad I. Feature selection using particle swarm optimization in intrusion detection. International Journal of Distributed Sensor Networks 2015;11(10):806954.

[38] Eesa AS, Orman Z, Brifcani AMA. A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems. Expert Systems with Applications 2015;42(5):2670–2679.

[39] Rai K, Devi MS, Guleria A. Decision tree based algorithm for intrusion detection. International Journal of Advanced Networking and Applications 2016;7(4):2828.

[40] 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.

[41] Bamakan SMH, Wang H, Yingjie T, Shi Y. An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization. Neurocomputing 2016;199:90–102.

[42] 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.

[43] Pajouh HH, Javidan R, Khayami R, Ali D, Choo KKR. A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks. IEEE Transactions on Emerging Topics in Computing 2016;.

[44] Shone N, Ngoc TN, Phai VD, Shi Q. A deep learning approach to network intrusion detection. IEEE Transactions on Emerging Topics in Computational Intelligence 2018;2(1):41–50.

[45] Naseer S, Saleem Y, Khalid S, Bashir MK, Han J, Iqbal MM, et al. Enhanced network anomaly detection based on deep neural networks. IEEE Access 2018;6:48231–48246.

[46] Sakr MM, Tawfeeq MA, El-Sisi AB. Network Intrusion Detection System based PSO-SVM for Cloud Computing. International Journal of Computer Network and Information Security 2019;11(3):22.

[47] Woo Jh, Song JY, Choi YJ. Performance Enhancement of Deep Neural Network Using Feature Selection and Preprocessing for Intrusion Detection. In: 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC) IEEE; 2019. p. 415–417.

[48] Cadima J, Cerdeira JO, Minhoto M. Computational aspects of algorithms for variable selection in the context of principal components. Computational statistics & data analysis 2004;47(2):225–236.

[49] Greselin F, Zitikis R. From the classical Gini index of income inequality to a new Zenga-type relative measure of risk: A modeller's perspective. Econometrics 2018;6(1):4.

[50] Goldberg DE, Holland JH. Genetic algorithms and machine learning 1988;.

[51] Bai Q. Analysis of particle swarm optimization algorithm. Computer and information science 2010;3(1):180.

[52] Kayacik HG, Zincir-Heywood AN, Heywood MI. Selecting features for intrusion detection: A feature relevance analysis on KDD 99 intrusion detection datasets. In: Proceedings of the third annual conference on privacy, security and trust, vol. 94; 2005. p. 1723–1722.

[53] Nykvist C, Larsson M, Sodhro AH, Gurtov A. A lightweight portable intrusion detection communication system for auditing applications. International Journal of Communication Systems 2020;p. e4327.

[54] Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of ICNN'95-International Conference on Neural Networks, vol. 4 IEEE; 1995. p. 1942–1948.

[55] Shi Y, Eberhart R. A modified particle swarm optimizer. In: 1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360) IEEE; 1998. p. 69–73.

[56] Ahmad I, e Amin F. Towards feature subset selection in intrusion detection. In: 2014 IEEE 7th Joint International Information Technology and Artificial Intelligence Conference IEEE; 2014. p. 68–73.

[57] Bre F, Gimenez JM, Fachinotti VD. Prediction of wind pressure coefficients on building surfaces using artificial neural networks. Energy and Buildings 2018;158:1429–1441.

[58] Quinlan J. Program for machine learning. C4 5 1993;.

[59] Kim G, Lee S, Kim S. A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. Expert Systems with Applications 2014;41(4):1690–1700.

[60] Mulay SA, Devale P, Garje G. Intrusion detection system using support vector machine and decision tree. International Journal of Computer Applications 2010;3(3):40–43.

[61] Shakil M, Fuad Yousif Mohammed A, Arul R, Bashir AK, Choi JK. A novel dynamic framework to detect DDoS in SDN using metaheuristic clustering. Transactions on Emerging Telecommunications Technologies 2019;p. e3622.

[62] Iwendi C, Khan S, Anajemba JH, Bashir AK, Noor F. Realizing an efficient IoMT-assisted patient diet recommendation system through machine learning model. IEEE Access 2020;8:28462–28474.

[63] Peng K, Leung V, Zheng L, Wang S, Huang C, Lin T. Intrusion detection system based on decision tree over big data in fog environment. Wireless Communications and Mobile Computing 2018;2018.

[64] Chung YY, Wahid N. A hybrid network intrusion detection system using simplified swarm optimization (SSO). Applied soft computing 2012;12(9):3014–3022.

[65] Tang TA, Mhamdi L, McLernon D, Zaidi SAR, Ghogho M. Deep learning approach for network intrusion detection in software defined networking. In: 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM) IEEE; 2016. p. 258–263.

[66] Dash T. A study on intrusion detection using neural networks trained with evolutionary algorithms. Soft Computing 2017;21(10):2687–2700.

Suleman Khan Suleman Khan received the master's degree from the Department of Computer Science, Air University Islamabad, in 2019. He is currently a Research Associate with Air University, Pakistan. His research interests include network security, machine learning, and data science.

Dr. Kashif Kifayat received his Ph.D. in Cyber Security from Liverpool John Moores University, Liverpool, UK, in 2008. He is currently working as Professor and Chair of Cyber Security Department at Air University, Islamabad, Pakistan. Prior to this, he was Reader in Cyber Security at Liverpool John Moores University, UK. His current research interests include network security, security of complex systems, intrusion detection, secure service composition, privacy-preserving data aggregation, cryptography, computer forensics and IoT security. He has published around 90 papers in international conference proceedings and journals and served in a number of conferences IPCs and journal editorial boards. He has also played a key role in many funded research and development projects related to his research topics.

Dr.Ali Kashif Bashir (Senior Member, IEEE) received the B.S. degree from the University of Management and Technology, Pakistan, the M.S. degree from Ajou University, South Korea, and the Ph.D. degree in computer science and engineering from Korea University, South Korea. He is currently a Senior Lecturer with the School of Computing, Mathematics, and Digital Technology, Manchester Metropolitan University, U.K. He is also a Distinguished Speaker of ACM. His past assignments include an Associate Professor of information and communication technologies with the Faculty of Science and Technology, University of the Faroe Islands, Denmark; Osaka University, Japan; the Nara National College of Technology, Japan; the National Fusion Research Institute, South Korea; Southern Power Company Ltd., South Korea; and the Seoul Metropolitan Government, South Korea. He is the author of over 80 peer-reviewed articles. He is supervising/co-supervising several graduate (M.S. and Ph.D.) students. His research interests include the Internet of Things, wireless networks, distributed systems, network/cyber security, and cloud/network function virtualization. Dr. Bashir has served as the Program Chair, the Publicity Chair, and the Track Chair on several conferences and workshops. He has delivered several invited and keynote talks, and reviewed the technology leading articles for journals like the IEEE Transactions on Industrial Informatics, the IEEE Communication Magazine, the IEEE Communication Letters, the IEEE Internet of Things, and the IEICE Journals, and conferences, such as the IEEE Infocom, the IEEE ICC, the IEEE Globecom, and the IEEE Cloud of Things. He is also serving as the Editor-in-Chief for the IEEE Future Directions Newsletter. He is also an Editor of several journals and also has served/serving as a Guest Editor on several special issues in journals of IEEE, Elsevier, and Springer.

Dr.Andrei Gurtov,(Senior Member, IEEE) is a Professor of Computer Science at Linköping University, Sweden. Previously he was at University of Oulu (3 years) and Aalto University (6 years) and visiting the International Computer Science Institute at Berkeley multiple times. He received his M.Sc (2000) and Ph.D. (2004) degrees in Computer Science from the University of Helsinki, Finland. Prof. Gurtov co-authored over 200 publications, including 4 books, 5 IETF RFCs, 6 patents, over 60 journal and 110 conference articles. He supervised 15 PhD theses. Professor Gurtov's research interests are in network protocols, security of vehicular, airborne, industrial systems, mobile, wireless and IoT networks, SmartGrids. He is an ACM Distinguished Scientist, IEEE ComSoc Distinguished Lecturer and Vice-chair of IEEE Sweden section. He received best paper awards at IEEE CSCN'17 and IEEE Globecom'11, was co-adviser of the best Doctoral Thesis in CS in Finland in 2017. He had served on numerous journal editorial boards and conference program committees, including IEEE Internet of Things journal, MDPI Sensors, IEEE ICNP, ACM MSWiM, and IFIP Networking. URL: http://gurtov.com.

Dr.Mehdi hassan has done his PhD in the area of intelligent disease diagnosis using medical imaging. He earned his PhD degree from PIEAS Pakistan in 2015. He has published several top rank international journal and conference papers. He has been working in Artificial Intelligence specifically deep neural networks, machine learning and image processing. He has supervised several MS students He is Co-PI in national center of excellence in cyber security lab. Currently, he is serving as Chair Department of Computer Science at Air University.