Multimedia Tools and Applications
https://doi.org/10.1007/s11042-020-09804-7

HFFPNN classifier: a hybrid approach for intrusion detection based OPSO and hybridization of feed forward neural network (FFNN) and probabilistic neural network (PNN)

T. Sree Kala[1] · A. Christy[2]

Received: 3 January 2020 / Revised: 6 August 2020 / Accepted: 2 September 2020
Published online: 16 October 2020

Abstract
Quick increase in web and system advancements has prompted significant increase in number of attacks and intrusions. Identification and prevention of these attacks has turned into an important part of security. Intrusion detection framework is one of the vital approaches to accomplish high security in computer systems and used to oppose attacks. Intrusion detection frameworks have reviled of dimensionality which tends to build time complexity and reduce resource use. Therefore, it is desirable that critical components of information must be examined by interruption detection framework to decrease dimensionality. These reduced features are then fed to a HFFPNN for training and testing on NSL-KDD dataset. HFFPNN is the hybridization of feed forward neural network (FFNN) and probabilistic neural network (PNN). Pre-processing of NSL-KDD dataset has been done to convert string attributes into numeric attributes before training. Comparisons with recent and relevant approaches are also tabled. Experimental results show the prominence of HFFPNN technique over the existing techniques in terms of intrusion detection classification. Therefore, the scope of this study has been expanded to encompass hybrid classifiers.

Keywords Feature reduction · Feature classification · Intrusion detection · HFFPNN · Feed-forward neural network (FFNN) · Probabilistic neural network (PNN) · Oppositional particle swarm optimization (OPSO)

✉ T. Sree Kala
sreekalatm@gmail.com

A. Christy
achristy@gmail.com

[1] Assistant Professor, Department of Computer Science, VISTAS, Chennai, India

[2] Department of Computer Science, Sathyabama University, Chennai, India

1 Introduction

Past few years have witnessed a growing recognition of intelligent techniques for the construction of efficient and reliable intrusion detection systems. An intrusion can be defined as "any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource" [11]. An Intrusion Detection System (IDS) provides an additional layer of security to network's perimeter defence, which is usually, implemented using a firewall. The goal of IDS is to collect information from a variety of systems and network sources, and then analyze the information for signs of intrusion and misuse. IDSs are implemented in hardware, software, or a combination of both [24]. An IDS provides monitoring and analysis of user and system activity, can audit system configuration and vulnerabilities, assess the integrity of critical system and data files, provide statistical analysis of activity patterns based on the matching with known attacks, analyze abnormal activity, and operate system audit [23]. One advantage of the IDS is its ability to document the intrusion or threat to an organization, thereby providing bases for informing the public regarding the latest attack patterns through system logs [13].

On the other hand, computers are under attacks and vulnerable to many threats. There is an increasing availability of tools and tricks for attacking and intruding networks [31]. An intrusion can be defined as any set of actions that threaten the security requirements (e.g., integrity, confidentiality, availability) of computer/network resource (e.g., user accounts, file systems, and system kernels). Intruders have promoted themselves and invented innovative tools that support various types of network attacks [33]. Hence, effective methods for intrusion detection (ID) have become an insisting need to protect our computers from intruders [28]. In general, there are two types of Intrusion Detection Systems (IDS); misuse detection systems and anomaly detection systems [14]. Anomaly detection systems with profile the normal behaviour of network or user or application and identifies deviations to these profiles which may be potential security breaches [6]. The second one is called misuse intrusion detection system, which uses attack signatures to compare with packet payloads for identifying intrusions [12]. Hence identifying new attacks is not possible using misuse detection whereas false alarm rates are more with respect to anomaly based detection [26]. Also, on the other side the speed, complexity and the size of the network is growing rapidly, especially when the network are open to public access, the number and type of intrusion increase dramatically making human analysis impossible [25].

This leads to the interest in using data mining techniques for network intrusion detection and analyzing various data sets such as flow data (network flow) [30]. There are various approaches that use data mining techniques such as, neural networks, SOM, SVM. Specific works related to the application of fuzzy logic, neural networks and agent based data mining approaches are discussed in [17]. There are mainly three data mining techniques that are widely applied in intrusion detection system which are clustering, association rule and sequential association rule [20]. In most IDS however, there is a high instances of false positives and false negatives which can be cumbersome to deal with for the network administrators. A false positive is an instance where an IDS incorrectly identifies a benign activity to be malicious while a false negative occurs when the IDS fails to detect a malicious activity [1]. During normal operation, IDS can generate thousands of false alarms per day [32]. Network intrusion detection systems - no matter if they are anomaly-based or signature-based - share a common problem: the high number of false alerts or false positives. The number of alerts collected by IDS can be up to 15,000 per day per sensor, and the number of false positives (FP) can be thousands per day [2]. These problems usually cause the final user, the security

manager to lose confidence in the alerts, lower the defence levels in order to reduce the number of false positives, or to have an overload of work to recognize true attacks due to IDS mistakes [19].

In this paper, we proposed effective hybrid approach for intrusion detection system using oppositional particle swarm optimization and probabilistic neural network using HFFPNN technique. HFFPNN is the hybridization of feed forward neural network (FFNN) and probabilistic neural network (PNN). Based on the optimization algorithm we select the optimal features. Here for feature extraction process oppositional particle swarm optimization algorithm (OPSO) is used [27]. The major contribution of the research for effective NSL-KDD dataset intrusion detection process are summarized as follow,

- The NSL-KDD dataset is obtained for pre-processing step to convert string attributes into numeric attributes before training process.
- An efficient optimization approach namely OPSO is done for selecting the optimal features, which has the advantages of achieving maximum accuracy of predicting features than the individual PSO algorithm and GA algorithm.
- Then HFFPNN technique is used for the classification of features. And also the proposed intrusion detection technique is experimented under various attacks.

The rest of the paper is organized as follows: a brief review of some of the literature works based on watermarking in relational database is presented in Section 2. In section 3 the background of the proposed method is explained in detail. The proposed technique for a hybrid approach for intrusion detection system using oppositional particle swarm optimization and probabilistic neural network using HFFPNN technique is given in section 4. The experimental results and the performance evaluation discussion are provided in Section 5. Finally, the conclusions are summed up in Section 6.

2 Literature survey

In recent times, intrusion detection has received a lot of interest among the researchers because it is widely applied for preserving the security within a network. Here, we present some of the techniques for intrusion detection system: Gao et al. [9] have presented a novel for network intrusion detection on cloud-based robotic system using fuzziness-based semi-supervised learning approach via ensemble learning (FSSLEL). First, they construct an ensemble system trained by the labeled data due to the good generalization ability of ensemble learning. Moreover, a fuzziness-based method is adopted for better utilizing the unlabeled data in data analysis. The noisy and redundant examples are removed by this way in the dataset. Finally, to combine both supervised and unsupervised parts they use the same ensemble approach. Moreover, Muamer N. Mohammad et al. [18] have introduced intrusion detection system by using intelligent data mining in weka environment, which is mainly focused on improved approach for Intrusion Detection System (IDS) based on combining data mining and expert system is presented and implemented in WEKA. The taxonomy consists of a classification of the detection principle as well as certain WEKA aspects of the intrusion detection system such as open-source data mining. The combining methods may give better performance of IDS systems, and make the detection more effective. Rather than, Ashfaqet al. [4] have presented a new fuzziness based semi-supervised learning technique, in order to enhance the performance of classifier for IDSs, using

unlabeled samples facilitated with supervised learning method. To throughput a fuzzy membership vector, single hidden layer feed forward neural network is trained and the sample classification on unlabeled samples is done by utilizing the fuzzy quantity.

Ahmed Youssef and Ahmed Emam [34] have examined network intrusion detection using data mining and network behaviour analysis, which is mainly focused on Traditional intrusion detection systems are limited and do not provide a complete solution for the problem. They search for potential malicious activities on network traffics; they sometimes succeed to find true security attacks and anomalies. However, in many cases, they fail to detect malicious behaviours (false negative) or they fire alarms when nothing wrong in the network (false positive). Krishna Kant et al. [29] have introduced intrusion detection using data mining techniques, which is mainly focused on data mining and soft computing techniques such as Artificial Neural Network (ANN), Support Vector Machine (SVM) and Multivariate Adaptive Regression Spline (MARS), etc. and the comparison shown between IDS data mining techniques and tuples used for intrusion detection. Either than, Solane Duque and Nizam bin Omar [8] have examined using data mining algorithms for developing a model for Intrusion Detection System (IDS), which is mainly focused on propose a model for Intrusion Detection System (IDS) with higher efficiency rate and low false positives and false negatives. K-means data mining algorithm followed by signature-based approach is proposed in order to lessen the false negative rate; and a system for automatically identifying the number of clusters may be developed. Moreover, Emil J. Khatib et al. [16] have introduced data mining for fuzzy diagnosis systems in LTE networks, which is mainly focused on Knowledge Base for a KBS from solved troubleshooting cases is proposed. This method is based on data mining techniques as opposed to the manual techniques currently used. The data mining problem of extracting knowledge out of LTE troubleshooting information can be considered a Big Data problem. Therefore, the proposed method has been designed so it can be easily scaled up to process a large volume of data with relatively low resources, as opposed to other existing algorithms.

Ambusaidi et al. [21] have built up a mutual information based algorithm that systematically chooses the optimal feature for classification. This mutual information based feature selection algorithm can deal with directly and nonlinearly subordinate information highlights. Its adequacy is assessed in the instances of system interruption recognition. An Intrusion Detection System (IDS), named Least Square Support Vector Machine based IDS (LSSVM-IDS), is developed utilizing the components chose by their proposed feature selection algorithm. The execution of LSSVM-IDS is assessed utilizing three interruption identification assessment datasets, to be specific KDD Cup 99, NSL-KDD and Kyoto 2006+ dataset.

The above works has been used for intrusion detection on cloud. This methods are produced the better results; Even though the security of cloud is not improved. So, to improve the cloud security, urgently new technique is needed. Therefore in this paper, an efficient intrusion detection system is proposed. The proposed approach is designed based on hybrid classifier.

3 Background information

3.1 FFNN model

The artificial neural network (ANN) was developed to estimate the intellectual learning procedures of organic neural systems [15]. A feed forward neural network (FFNN) is a kind of ANN in which the neuron associations don't frame a cycle and the data just pushes ahead

from the input hubs through the output hubs to the yield hubs with no loops or cycles; as it were, data just goes from the input layer to the main hidden layer, at that point proceeds onward to the second hidden layer and so on, finally flowing out the output layer.

The neuron is the essential component in the FFNN; if data from m neurons in the previous layer stream into one neuron i, in numerical terms, the data stream outs as an ensemble, as given in Eq. (1):

$$z_i = a\left(\sum_{j=1}^{m} t_{ij}c_{ij} - d_{i0}\right) \tag{1}$$

where t_{ij} is the weight of the connection from neuron j of the previous layer to the present neuron i, c_{ij} is the relating data, and d_{i0} is the inherent threshold for neuron i which is dealt with as an ordinary weight with the input data being −1.

Also, 'a' is the transfer or activation function for which the linear function $a(c) = c$, logistic sigmoid function $a(c) = 1/(1 + e^{-c})$ and hyperbolic digression work $a(c) = 2/(1 + e^{-2c}) - 1$ are generally utilized. Figure 1 demonstrates a general structure of the multi-hidden layer FFNN. If there are m and n neurons in the input layer and the output layer, the FFNN can map a point in the $\Re^m$ space to the $\Re^n$ space, which is like nonlinear regression [3].

It is most important to have weights that precisely reflect the connection between the input and output factors to develop a FFNN. Given a training data set with P input-output vector combines, the most effective strategy for tackling this issue is the back-propagation algorithm [26, 34] which can limit the execution work, as appeared in Eq. (2)

$$E = \frac{1}{2Z}\sum_{z=1}^{Z}\sum_{g=1}^{G}\left(f_g^z - h_g^z\right)^2 \tag{2}$$

This execution work result in the global mean sum squared error between the calculated outputs $h^{(z)}$ and the targeted outputs $f^{(z)}$, for which z and g are the records for the z^{th} training sample also, for the g^{th} part of the output vector.

3.2 PNN model

The probabilistic neural network (PNN) is a Bayes–Parzen classifier [5] that is often an excellent pattern classifier in practice. The foundation of the approach is well known decades ago in 1960s; however, the method was not of a widespread use because of the lack of sufficient computation power. Then the Bayes–Parzen classifier could be broken up into a large number of simple processes implemented in a multilayer neural network each of which could be run independently in parallel.

The probabilistic neural network is primarily based on Bayes–Parzen classification; it is of interest to discuss briefly both Bayes theorem for conditional probability and Parzen's method for estimating probability density function of random variables. In order to understand Bayes' theorem, consider a sample $a = [a_1, a_2, .. \ldots, a_n]$ taken from a collection of samples belonging to a number of distinct populations 1, 2, …, p. …, P. Assuming that the (prior) probability that a sample belongs to the pth population is l_p, the cost associated with misclassifying that sample is s_p, and that the true probability density function of all populations $f_1(a)$, $f_2(a)$, …, $f_p(a)$, …, $f_P(a)$ are known, Bayes theorem classifies an unknown sample into the ith population if;

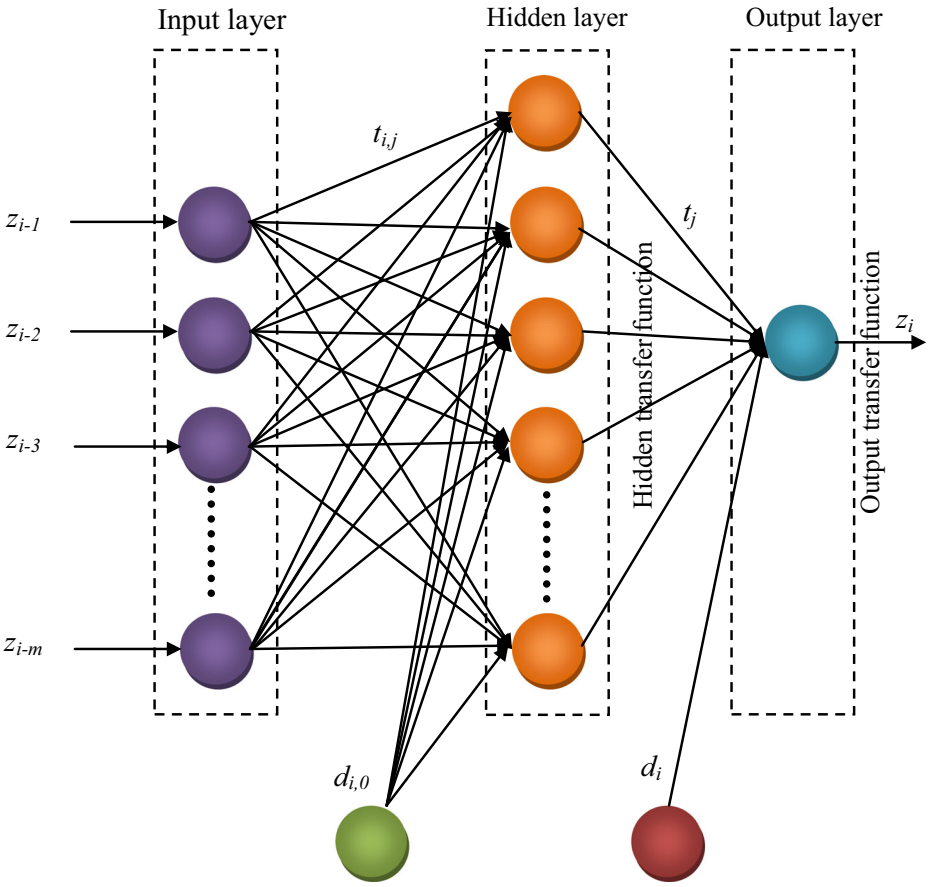

Fig. 1 Feed Forward Neural Network model

$$l_i s_i f_i(a) > l_k s_k f_k(a) \forall k \neq i, k = 1, 2, \ldots, P \tag{3}$$

The density function $f_p(a)$ corresponds to the concentration of class p examples around the unknown example. As seen from Eq. (3), Bayes' theorem favors a class that has high density approximately the unknown sample, or if the cost of misclassification or prior probability is high.

The biggest problem with the Bayes' classification approach lies in the fact that the probability density function $f_p(a)$ is not usually known. In nearly all standard statistical classification algorithms, some knowledge regarding the underlying distribution of the population of all random variables used in classification should be known or reasonably assumed. Most often, normal (Gaussian) distribution is assumed; however, the assumption of normality cannot always be safely justified. When the distribution is not known (which is often the case) and the true distribution deviates considerably from the assumed one, the traditional statistical methods normally run into major

classification problems resulting in high misclassification rate. There is a need to derive an estimate off $p(a)$, from the training set composed of the training example, rather than just assume normal distribution. The resulting distribution will be a multivariate probability density function (PDF) that combines all the explanatory random variables. The multivariate PDF estimator, $q(x)$, may be expressed as:

$$q(a_1, a_2, \ldots, a_n) = \frac{1}{S\omega_1, \omega_2, \ldots, \omega_n} \sum_{i=1}^{S} G\left(\frac{a_1 - a_{1-i}}{\omega_1}, \frac{a_2 - a_{2-i}}{\omega_2}, \ldots, \frac{a_n - a_{n-i}}{\omega_n}\right) \quad (4)$$

where $\omega_1, \omega_2, \ldots, \omega_n$ are the smoothing parameters representing standard deviation (also called window or kernel width) around the mean of n random variables $a_1, a_2, \ldots, a_n$, G is a weighting function to be selected with specific characteristics and S is the total number of training examples. If all smoothing parameters are assumed equal such as $\omega_1 = \omega_2 = . \ldots = \omega_n = \omega$ and a bell-shaped Gaussian function is used for G, a reduced form of Eq. (4) is as follows [10]:

$$q(a) = \frac{1}{(2\pi)^{\pi/2}\omega^n} \times \frac{1}{S} \sum_{i=1}^{S} \exp\left[-\frac{\|(a - a_i)\|^2}{2\omega^2}\right] \quad (5)$$

where a the vector of random variables and a_i is the ith training vector. Equation (5) represents the average of the multivariate distributions where each distribution is centered at one distinct training example. It is worth mentioning that the assumption of a Gaussian weighting function does not imply that the overall PDF will be Gaussian (normal), however, other weighting functions such as the reciprocal function $g(c) = 1/1 + c^2$ may be used. As the sample size, S increases, the Parzen's PDF estimator asymptotically approaches the true underlying density function.

Regarding the network's operation based on the a fore mentioned mathematics, consider the simple network architecture in Fig. 2 with n input nodes in the input layer, two population

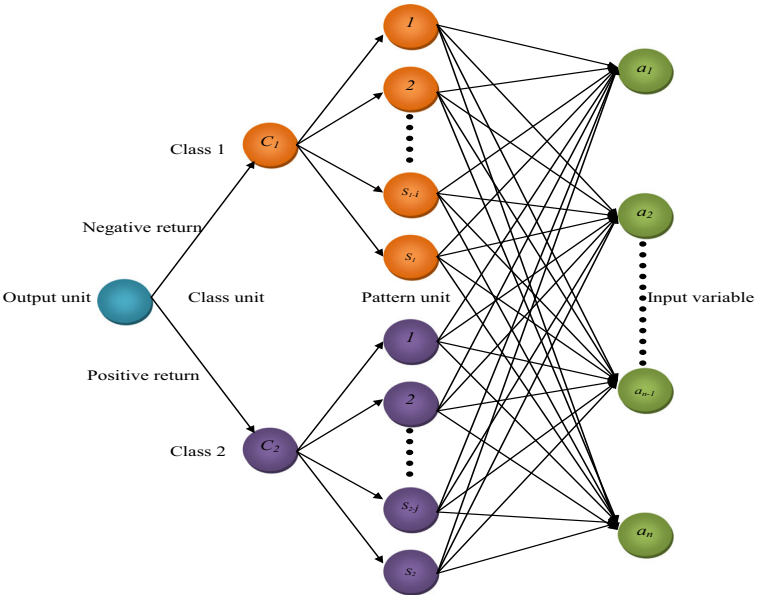

Fig. 2 Probabilistic Neural Network model

classes (classes 1 and 2), S_1 training examples belonging to class 1, and S_2 examples in class 2. The pattern layer is designed to contain one neuron for each training case available and the neurons are split into two classes. The summation layer contains one neuron for each class. The output layer contains one neuron that operates trivial threshold discrimination; it simply retains the maximum of the two summation neurons. This procedure prevents any bias in the network to the correctly classified examples, and thus will be followed in this study.

4 Proposed intrusion detection using HFFPNN

The primary intention of this paper is to design and develop a technique for intrusion detection system. The intrusion detection system (IDS) is one of the most important components of a network management system to prevent attacks from paralyzing the entire network. However, detecting the new type of attacks on a network system is a very difficult problem from the perspective of the classification mechanism of IDS. According to this work, an intrusion detection system based on OPSO with HFFPNN to address the classification problem in system contains two modules namely feature selection module and classification module. In feature selection module, the important feature will be selected with the use of OPSO. In classification module, the selected features will be taken for training using HFFPNN. Subsequently, test data will be given to the testing network, which outputs if the data is intruded or not. The overview of the proposed approach is shown in Fig. 3.

4.1 Data pre-processing

The suggested technique uses the text data as the input, initially the input text data is fed to the preprocessing stage. In preprocessing, at the outset, it reads the input text data and then converts the string attributes to numeric attributes from the input text data. The steps involved in data preprocessing is shown in Fig. 4.

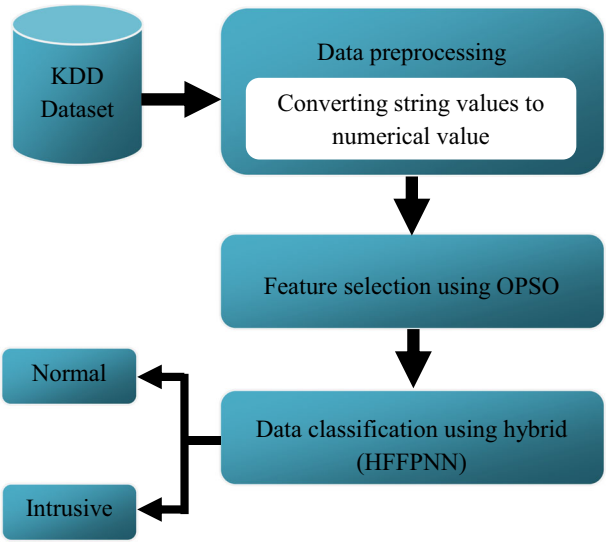

Fig. 3 Overview of proposed approach

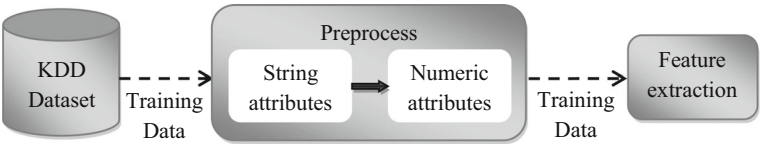

Fig. 4 Data preprocessing steps

4.2 Feature selection using OPSO

After the preprocessing, the input data are given to the feature selection process. The large number of features is the great obstacles for classification. So, the important features are selected in this paper. For feature selection process OPSO algorithm is Shown in Fig. 5. OPSO is a combination of oppositional based learning (OBL) [22] and PSO. Particle swarm

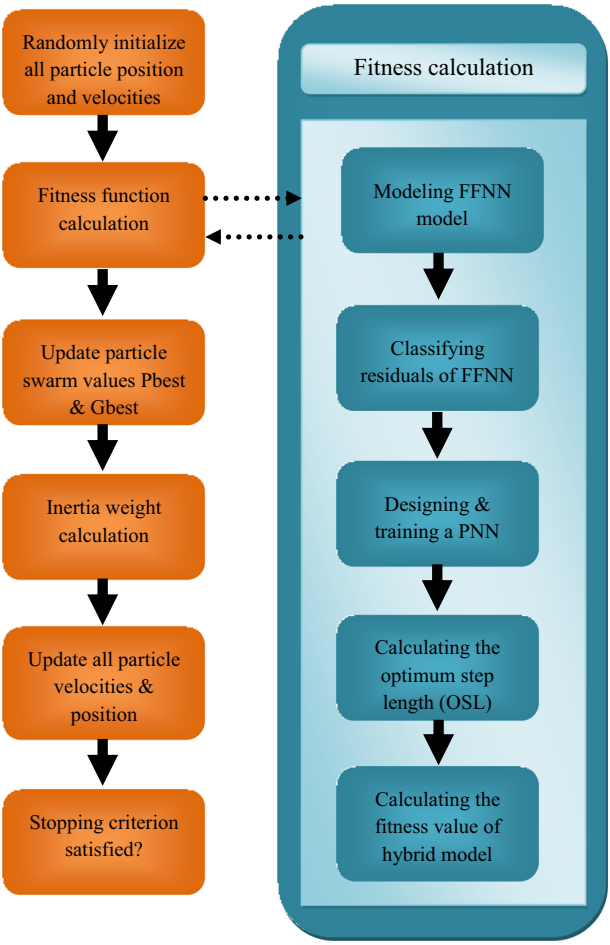

Fig. 5 Proposed OPSO-HFFPNN process

optimization is a population-based optimization algorithm which is inspired based on the concept of birds. The step by step process of feature selection is explained below;

Step 1: **Initialization:** Solution initialization is an important feature for all optimization algorithms. In this section, initially the solutions are assigned randomly. The sample solution format is given in below Table 1:

Step 2: **Opposite solution generation:** After the solution generation process, the opposite solutions are generated. The opposite solution is calculated using Eq. (6).

$$OW_i = [ow_i1, \ldots\ldots, ow_iC_n] \tag{6}$$

where $ow_i = Low_i + Up_i - w_i$ with $ow_i \in [Low_i, Up_i]$ is the position of i^{th} opposite agent OW_i in the d^{th} dimension of oppositional population.

Step 3: **Fitness calculation:** After the solution initialization, the fitness of each solution is calculated. The fitness calculation is given in Eq. (7).

$$Fitness\ function = Maximum\ Accuracy \tag{7}$$

Step 4: **Updation using PSO:** After fitness calculation, each solution is updated. For updation, each solution position and velocity are updated.

$$V_i^{new} = V_i + \gamma_1.a_1.(pbest_i - P_i) + \gamma_2.a_2.(gbest_i - P_i) \tag{8}$$

$$P_i^{new} = P_i + V_i^{new} \tag{9}$$

where,

i Weight.
γ_1, γ_2 Learning rates governing the weight towards its best position.
a_1, a_2 Random numbers that are uniformly distributed in the range [0, 1].

Table 1 Sample solution format

Records	F_1	F_2	F_3	....	F_{41}
R_1	1	1	0	....	1
R_2	0	1	1	....	0
R_3	1	0	0	....	1
....	....	....	....	....	....
R_n	1	0	1		0

V_i indicates the current velocity.

Step 5: **Termination criteria:** Above operations are continued until finding the optimal features. Once the optimal feature is obtained, then the algorithm will be terminated. The selected feature is given to the classification process.

4.3 Intrusion detection using HFFPNN model

Despite the numerous time series models available, the accuracy of time series prediction currently is fundamental to many decision processes, and hence, never research into ways of improving the effectiveness of prediction models been given up. In the literature, different combination techniques have been proposed in order to overcome the deficiencies of single models and yield more accurate hybrid models. In this paper, in contrast of the traditional hybrid techniques, which combine different time series models together, a time series model FFNN are combined with a classifier PNN model. The aim of this proposed model is to use the unique advantages of the probabilistic neural networks as classifier models in order to classify and determine the existing trend in the residuals of the FFNN. The procedure of the proposed model can be summarized in the five stages shows in Fig. 6. In the first stage, the under-study time series (z_e) is initially modelled by a FFNN, as follows.

$$z_e = F_{FFNN}(e) + r_e \tag{10}$$

where $F_{FFNN}(e)$ and r_e are the estimated values and residuals of the FFNN at time period e, respectively.

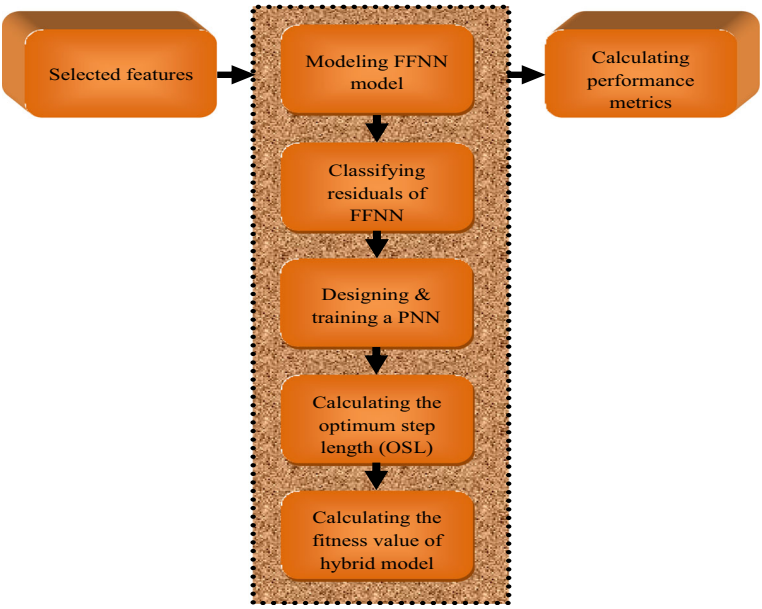

Fig. 6 Process involved in feature classification

4.3.1 Class of residual

In the second stage, according to the obtained results of the first stage, the estimated values and residuals of the feed forward neural network model and desired level of error (DLE), the residuals of feed-forward neural network are classified in three categories as follows. The desired level of error is a non-negative value that determines the sensitivity of the proposed model against the residuals of the FFNN. The DLE value is often the ideal level of accuracy for under-study problem, which is chosen by decision maker.

(i) The residuals, which are greater than the desired level of error ($DLE < r_i/r_i$), are classified in category one with assigned number "trend = 1".
(ii) The residuals, which are less than the negative of the desired level of error ($-DLE > r_i/r_i$), are classified in category two with assigned number "trend = −1".
(iii) The residuals, which are less than or equal to the desired level of error or are greater than or equal to the negative of the desired level of error $DLE \geq |r_i|$, are classified in category three with assigned number "trend = 0".

4.3.2 Optimum step length (OSL)

In the third stage, a classifier model is applied in order to distinguish the existing trend in the residuals. In this paper, PNN is used as classifier. Technically, PNN is able to deduce the class/group of a given input vector after the training process is completed. There are a number of appealing features, which justify our adoption of this type of neural networks in this study. First, training of probabilistic neural networks is rapid, enabling us to develop a frequently updated training scheme. Essentially, the network is re-trained each time the data set is updated and thus the most current information can be reflected in estimation. Second, the logic of PNN is able to extenuate the effects of outliers and questionable data points and thereby reduces extra effort on scrutinizing training data. Third and the most important, PNN are conceptually built on the Bayesian method of classification which given enough data, is capable of classifying a sample with the maximum probability of success.

The PNN is designed and trained by considering the assigned numbers of each category and subset of effective variables as output and input values, respectively. The effective variables on the target value of the mentioned PNN at time period e are as follows:

(i) Lags 1until $x^{th}$of the under-study time series at time period $e(z_{e-1}, z_{e-2}, \ldots, z_{e-x})$.
(ii) Lags1untily^{th} of the FFNN residuals at time period $e(r_{e-1}, r_{e-2}, \ldots, r_{e-y})$.
(iii) Estimated value of the FFNN at time period $e(\widehat{z}_e)$.
(iv) Lags 1untilw^{th} of the estimated values of the FFNN at time $e(\widehat{z}_{e-1}, \widehat{z}_{e-2}, \ldots, \widehat{z}_{e-w})$.

where *x, y, w* are integer. In fourth stage, according to the obtained results of the previous stages, the target values obtained from the designed PNN (−1, 0, +1) and estimated values of the FFNN OSL is calculated using a mathematical programming model as follows:

$$\begin{aligned} &Minimize\ u = \sum_{e=1}^{n} v_e \\ &Subject\ to \begin{cases} v_e \geq z_e - \widehat{z}_e - T_g(e) \times OLS & for\ e = 1, 2, \ldots, n \\ v_e \geq \widehat{z}_e - z_e - T_g(e) \times OLS & for\ e = 1, 2, \ldots, n \\ y_e v_e \leq y_e z_e - y_e \widehat{z}_e - y_e T_g(e) \times OLS & for\ e = 1, 2, \ldots, n \\ (1-y_e)v_e \leq (1-y_e)z_e - (1-y_e)\widehat{z}_t - (1-y_e)T_g(e) \times OLS & for\ e = 1, 2, \ldots, n \\ OSL, v_e \geq 0, y_e \in \{0, 1\} \quad e = 1, 2, \ldots, n \end{cases} \end{aligned} \tag{11}$$

where $T_g(e)$ the target value is obtained from PNN at time period e and n is the training sample size. In the fifth stage, according to the obtained results of the previous stages the estimated values of FFNN, target values of PNN, and the OSL the fitted values of the proposed model is calculated as follows:

$$F_x(e) = F_{FFNN}(e) + \left(T_g(e) \times OSL\right) \tag{12}$$

where $F_x(e)$ *and* $F_{FFNN}(e)$ are the fitted values of the proposed model and FFNN model at time period e respectively.

5 Experimental result and discussion

The results talked about in this section were acquired from the proposed strategy implemented in a system with the accompanying details: CPU Intel® Pentium 1.9 GHz, 64-bit operating system, Microsoft® Windows 10, 4 GB of RAM. A hybrid approach for intrusion detection system using oppositional particle swarm optimization and probabilistic neural network is implemented in JAVA and the experimentation is carried out on NSL-KDD dataset.

5.1 Dataset description

The improved version of KDD cup 99 dataset is referred as the NSL-KDD dataset [7]. The NSL-KDD dataset is comprised of a lot of information. Numerous specialists performed different investigations on NSL-KDD dataset and executed different tools and systems. In any case, their regular point was to develop effective IDS. A point by point NSL-KDD dataset execution utilizing distinctive machine learning procedures was performed with the utilization of a WEKA device and talked about in [21]. In dataset, each record has 41 attributes representing to various stream features. Each sample is marked either normal or attack sort. Beside normal data, records that compare to the 38 different attack types are found in the NSL-KDD dataset that are shown in Table 3.

5.2 Quality metrics

To evaluating the clustering performance, the proposed method uses different types of measures such as Sensitivity, specificity, accuracy, false positive rate (FPR) and false negative rate (FNR).

5.2.1 Sensitivity

The ratio of a number of true positives to the sum of true positive and false negative is called as sensitivity.

$$Sensitivity = \frac{Count\ of\ TP}{Count\ of\ TP + Count\ of\ FN} \times 100 \tag{13}$$

5.2.2 Specificity

Specificity is defined as the ratio of a number of true negatives to the sum of true negatives and false positives.

$$Specificity = \frac{Count\ of\ TN}{Count\ of\ TN + Count\ of\ FP} \times 100 \tag{14}$$

5.2.3 Accuracy

Accuracy can be calculated using the measures of sensitivity and specificity. It is denoted as follows,

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \tag{15}$$

where,

TP True Positive.
TN True Negative.
FP False Positive.
FN False Negative.

5.2.4 False positive rate (FPR)

FPR propose the division of persons, who were incorrectly classified as positive, but really be a member of negative categorization.

$$FPR = \frac{FP}{FP + TN} \tag{16}$$

Table 2 Feature reduced dataset by using OPSO algorithm

Dataset	Total Features	Reduced Features	Selected features
Feature reduced dataset	41	22	1, 4, 5, 6, 7, 9, 10, 11, 12, 16, 17, 24, 26, 28, 29, 30, 31, 32, 34, 37, 39, 41.

Table 3 Detection result obtained by the hybrid approach for the new attacks

Attack name	Instance number in the test set	Instance number detected by the hybrid approach
'apache2'	737	593
'back'	359	285
'buffer_overflow'	20	18
'ftp_write'	3	3
'guess_passwd'	1231	969
'httptunnel'	133	109
'imap'	1	1
'ipsweep'	141	115
'land'	7	5
'loadmodule'	2	2
'mailbomb'	293	250
'mscan'	996	791
'multihop'	18	15
'named'	17	12
'neptune'	4657	3663
'nmap'	73	60
'normal'	9710	7681
'perl'	2	2
'phf'	2	2
'pod'	41	33
'portsweep'	157	119
'processtable'	685	542
'ps'	15	12
'rootkit'	13	10
'saint'	319	261
'satan'	735	579
'sendmail'	14	11
'smurf'	665	539
'snmpgetattack'	178	140
'snmpguess'	331	263
'sqlattack'	2	2
'teardrop'	12	9
'udpstorm'	2	1
'warezmaster'	944	763
'worm'	2	2
'xlock'	9	9
'xsnoop'	4	3
'xterm'	13	12

5.2.5 False negative rate (FNR)

FNR is where analysis outcome specify as negative, but really be a member of positive categorization.

$$FNR = \frac{FN}{FN + TP} \tag{17}$$

5.3 Comparative analysis

To prove the effectiveness of proposed method, the proposed OPO + HFFNN is compared with different methods.

Table 4 Attacks used in proposed OPSO-HFFPNN

Attacks	Sensitivity	Specificity	Accuracy	FPR	FNR
'apache2'	80.4613	100	99.3612	0.047	19.5387
'back'	79.3872	100	99.6717	0.014	20.6128
'buffer_overflow'	90	100	99.9911	0.011	10
'ftp_write'	100	100	100	0.045	0
'guess_passwd'	78.7165	100	98.8378	0.014	21.2835
'httptunnel'	81.9549	100	99.8935	0.042	18.0451
'imap'	100	100	100	0.026	0
'ipsweep'	81.5603	100	99.8847	0.016	18.4397
'land'	71.4286	100	99.9911	0.041	28.5714
'loadmodule'	100	100	100	0.025	0
'mailbomb'	85.3242	100	99.8093	0.034	14.6758
'mscan'	79.4177	100	99.0906	0.031	20.5823
'multihop'	83.3333	100	99.9867	0.016	16.6667
'named'	70.5882	100	99.9778	0.033	29.4118
'neptune'	78.6558	100	95.5906	0.035	21.3442
'nmap'	82.1918	100	99.9423	0.043	17.8082
'normal'	79.104	100	90.9994	0.042	20.896
'perl'	100	100	100	0.016	0
'phf'	100	100	100	0.00034	0
'pod'	80.4878	100	99.9645	0.00026	19.5122
'portsweep'	75.7962	100	99.8314	0.029	24.2038
'processtable'	79.1241	100	99.3657	0.02	20.8759
'ps'	80	100	99.9867	0.034	20
'rootkit'	76.9231	100	99.9867	0.038	23.0769
'saint'	81.8182	100	99.7427	0.029	18.1818
'satan'	78.7755	100	99.308	0.045	21.2245
'sendmail'	78.5714	100	99.9867	0.02	21.4286
'smurf'	81.0526	100	99.4411	0.03	18.9474
'snmpgetattack'	78.6517	100	99.8314	0.04	21.3483
'snmpguess'	79.4562	100	99.6984	0.016	20.5438
'sqlattack'	100	100	100	0.029	0
'teardrop'	75	100	99.9867	0.016	25
'udpstorm'	50	100	99.9956	0.039	50
'warezmaster'	80.8263	100	99.1971	0.026	19.1737
'worm'	100	100	100	0.03	0
'xlock'	100	100	100	0.021	0
'xsnoop'	75	100	99.9956	0.039	25
'xterm'	92.3077	100	99.9956	0.026	7.6923

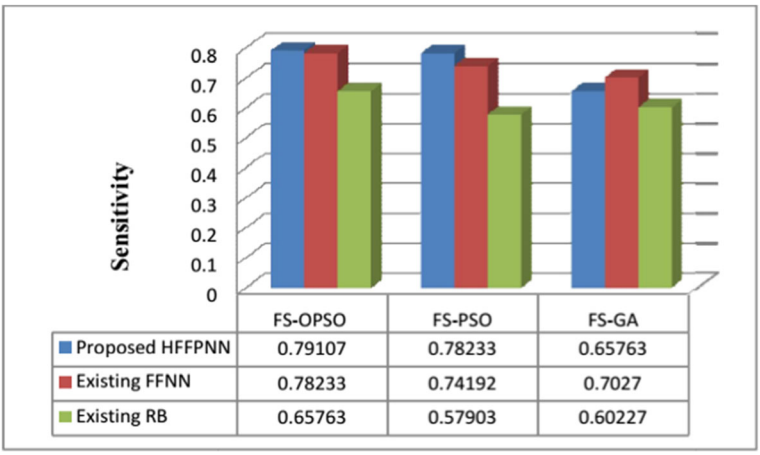

Fig. 7 Sensitivity plot for proposed vs. existing techniques by varying feature selection technique

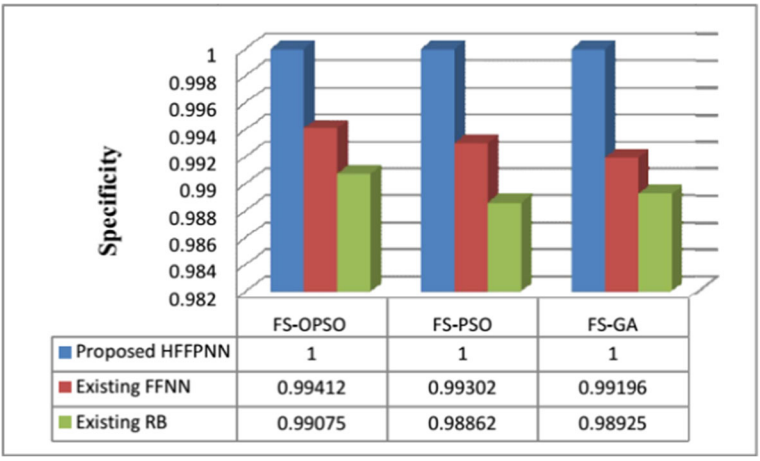

Fig. 8 Specificity plot for proposed vs. existing techniques by varying training percentage of data

For NSL-KDD data, there are 38 new attacks are used in the test set, relating to 9710 instances, that don't show up in the 10% training dataset. For these 38 attacks, in Table 2 the detection results obtained by the hybrid approach are recorded. By analyzing Table 3, the test set instance of each attack has been reduced when instance number detected by the hybrid approach. Various attacks were used to analyze the performance of the proposed OPSO-HFFPNN technique. The detection results obtained by the hybrid approach with different test set are shown in Table 4. From Table 4 there are large changes can be found in terms of sensitivity, specificity, accuracy, FPR and FNR. It is noted that the sensitivity, specificity, FPR and FNR values of the proposed method is high when the OPSO algorithm is used feature selection. Finally, all the performance metrics values obtained for proposed and the existing techniques on varying feature selection techniques are plotted graphically in the below Figs. 7, 8, 9, 10 and 11.

The sensitivity plot of proposed and existing techniques by varying feature selection (FS) technique is shown in Fig. 7. By analyzing the graph the proposed sensitivity value is high when comparing with existing techniques. The average sensitivity value of existing FS-PSO is 0.70109

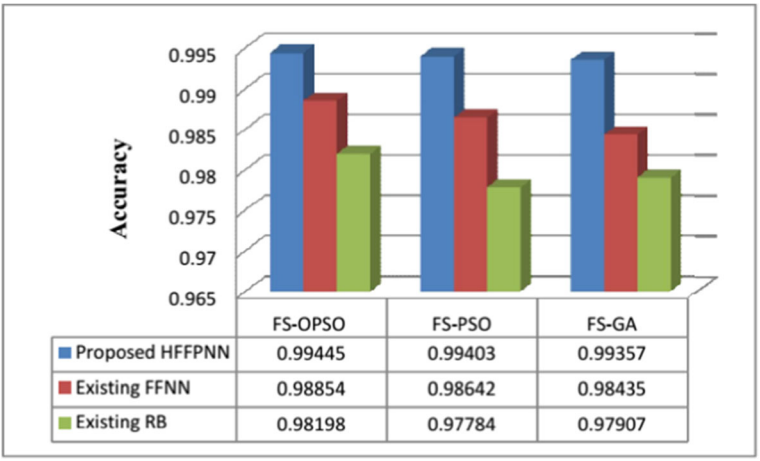

Fig. 9 Accuracy plot for proposed vs. existing techniques by varying feature selection technique

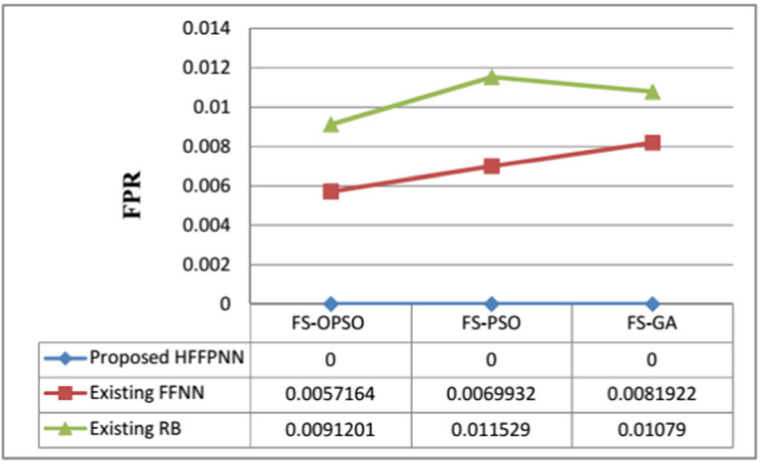

Fig. 10 FPR plot for proposed vs. existing techniques by varying feature selection technique

and FS-GA is 0.6542 while the proposed FS-OPSO achieves 0.743677 which is greater than the existing techniques. Hence the proposed sensitivity value is better than the existing feature selection techniques. The specificity plot of proposed and existing techniques by varying feature selection technique is shown in Fig. 8. By analyzing the graph the proposed specificity value is very high when compared with existing techniques. The average specificity value of existing FS-PSO is 0.99388 and FS-GA is 0.993737 while the proposed FS-OPSO achieves 0.994957 which is greater than the existing techniques. Hence the proposed specificity value is better than the existing feature selection techniques. The accuracy plot of proposed and existing techniques by varying feature selection technique is shown in Fig. 9. By analyzing the graph the proposed accuracy value is very high when compared with existing techniques. The average accuracy value of existing FS-PSO is 0.986097 and FS-GA is 0.985663 while the proposed FS-OPSO achieves 0.988323 which is greater than the existing techniques. Hence the proposed accuracy value is better than the existing feature selection techniques.

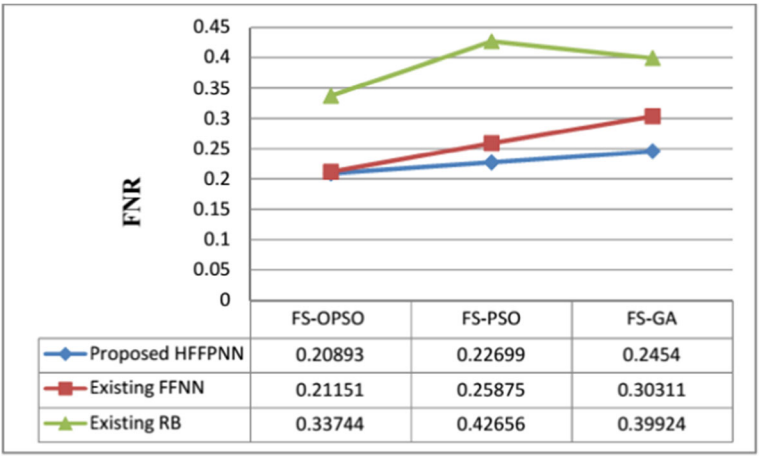

Fig. 11 FNR plot for proposed vs. existing techniques by varying feature selection technique

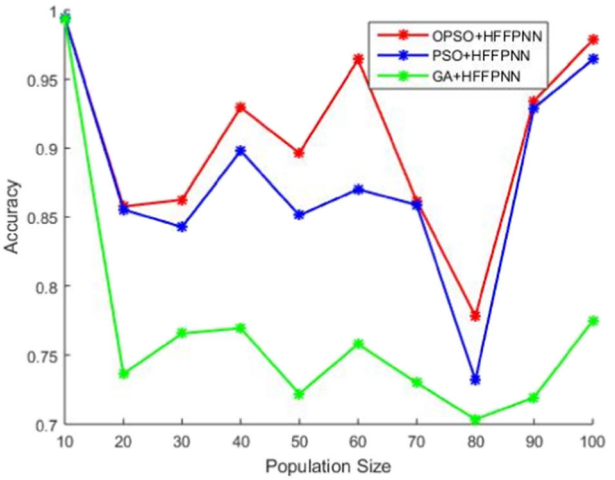

Fig. 12 Accuracy value feature classification by varying population size

Table 5 Comparative analysis of performance measures using existing PSO with existing and proposed feature classification techniques

	Sensitivity	Specificity	Accuracy	FPR	FNR
PSO-NN	74.17	99.30	98.64	69	25.82
PSO-RB	57.95	98.86	97.78	1.13	42.04
PSO-HFFPNN	77.80	1	99.41	0	22.19

Table 6 Comparative analysis of performance measures using existing GA with existing and proposed feature classification techniques

	Sensitivity	Specificity	Accuracy	FPR	FNR
GA-NN	70.32	99.19	98.43	80	29.67
GA-RB	60.16	98.92	97.90	1.07	39.83
GA-HFFPNN	75.72	1	99.36	0	24.27

Table 7 Comparative analysis of performance measures using proposed OPSO with existing and proposed feature classification techniques

	Sensitivity	Specificity	Accuracy	FPR	FNR
OPSO-NN	78.13	99.40	98.84	59	21.86
OPSO-RB	66.18	99.08	98.22	91	33.81
OPSO-HFFPNN	78.88	1	99.44	0	21.11

The FPR plot of proposed and existing techniques by varying feature selection technique is shown in Fig. 10. The FPR value must be low for attaining the better performance. By analyzing the graph the proposed FPR value is very low when compared with existing techniques. In our proposed FS-OPSO technique the FPR value is 0. Hence the proposed FPR value is better than the existing feature selection techniques. The FNR plot of proposed and existing techniques by varying feature selection technique is shown in Fig. 11. The FNR value must be low to get the better performance. By analyzing the graph the proposed accuracy value is low when compared with existing techniques. The average FNR value of proposed FS-OPSO is 0.252627 while the existing FS-PSO gets 0.3041 and FS-GA value is 0.395917. Hence the proposed FNR value is better than the existing feature selection techniques. From the below figures, it is clear that the evaluation metrics outcomes of the proposed approach are better than the existing approaches using OPSO technique.

The performance assessment of the proposed intrusion detection system using HFFPNN classification method is shown in this section with various existing methods. The analysis is made in the basis of varying various feature selection techniques.

The accuracy value plays the major factor in intrusion detection system. It is important for the method to provide the high accuracy value in order to serve as the best method and the Fig. 12 shows the comparative analysis for accuracy value by varying the population size. From the graph it is clear that the accuracy value for the existing PSO-HFFPNN method is 0.87958 and GA-HFFPNN method is 0.76716 while for the proposed OPSO-HFFPNN method achieves 0.90569. Since the accuracy value for the proposed method is very high it seems to be better than the existing methods.

Moreover, the performance measures such as Accuracy, sensitivity, specificity, FPR and FNR values of the feature classification techniques is compared with existing PSO and GA with the proposed OPSO algorithm form Tables 5, 6 and 7. By comparing Tables 5, 6 and 7, the proposed OPSO approach gives the better feature classification execution result than the existing PSO and GA techniques. Therefore, it can be concluded that in general, the OPSO-HFFPNN method has a satisfactory performance for the intrusion detection system conditions.

6 Conclusion

The study proposed a new intelligent intrusion detection framework that deals with reduced number of features. In order to reduce time complexity and to improve resource utilization, the pre-processing method is used to convert string attributes into numerical attributes from the dataset. Feature from the dataset is reduced by using OPSO technique. A classification system was designed by HFFPNN which was trained on NSL-KDD dataset. Here 38 different attacks were used and tested for performance evaluation. To verify the intrusion detection capability of the proposed weighted OPSO-HFFPNN model, the PSO-HFFPNN, PSO-FFNN, PSO-RA, GA-HFFPNN, GA-FFNN, GA-RB, OPSO-FFNN and OPSO-RB models were employed for comparison. Sensitivity, specificity, accuracy, FPR and FNR were utilized as performance indicators to survey the prediction performance of the proposed OPSO-HFFPNN model. It was found that the OPSO-HFFPNN model had the best performance of all the prediction models, as in each test case, the FPR and FNR were the lowest. Therefore, the empirical results obtained from the experiments show that HFFPNN perform better than the existing techniques with respect to various performance metrics.

References

1. Ahmad I, Abdullah AB, Algamdi AS (2009) Application of artificial neural network in detection of probing attacks. IEEE Symposium on Industrial Electronics and Applications (ISIEA 2009), October 4-6
2. Aljawarneh S, Aldwairi M, Yassein MB (2017) Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. J Comput Sci
3. An N, Zhao W, Wang J, Shang D, Zhao E (2013) Using multi-output feedforwardneural network with empirical mode decomposition based signal filtering forelectricity demand forecasting. Energy 49:279–288
4. Ashfaq RAR, Wang X-Z, Huang JZ, Abbas H, He Y-L (2017) Fuzziness based semi-supervised learning approach for intrusion detection system. Inf Sci 378:484–497
5. Berthold MR, Diamond J (1998) Constructive training of probabilistic neural networks. Neurocomputing 19(1):167–183
6. Conti G, Abdullah K, Grizzard J, Stasko J, Copeland JA, Ahamad M, Owen HL, Lee C (2006) Countering security information overload through alert and packet visualization, Published by the IEEE Computer Society
7. Dhanabal L, Shantharajah SP (2015) A Study on NSL-KDD dataset for intrusion detection system based on classification algorithms. Int J Adv Res Comp Commun Eng 4:446–452
8. Duquea S, Omar MN b (2015) Using data mining algorithms for developing a model for Intrusion Detection System (IDS). Proced Comp Sci 61:46–51
9. Gao YL, Jin Y, Chen Y, Wu J (2018) A novel semi-supervised learning approach for network intrusion detection on cloud-based robotic system, IEEE Access, 1–12
10. Ge SS, Yang Y, Lee TH (2008) Hand gesture recognition and tracking based on distributed locally linear embedding. Image Vis Comput 26:1607–1620
11. Heady R, Luger G, Maccabe A, Servilla (1990) The architecture of a network level intrusion detection system, Technical report, Computer Science Department, University of New Mexico
12. Hooper E, Egham (2007) an intelligent intrusion detection and response system using hybrid ward hierarchical clustering analysis, IEEE International Conference on Multimedia and Ubiquitous Engineering
13. Jiawei H, Micheline K (2006) Data Mining Concepts and techniques", second edition, China Machine Press, pp. 296–303
14. Jin H, Sun J, Chen H, Han Z (2004) A fuzzy data mining based intrusion detection model, Proceedings of the 10th IEEE International Workshop on Future Trends of Distributed Computing Systems
15. Jun W, Lingyu T, Yuyan L, Peng G (2017) A weighted EMD-based prediction model based on TOPSIS and feed forward neural network for noised time series. Knowl-Based Syst 132:167–178
16. Khatib EJ, BarcoR, Gómez-Andrades A (2015) Data mining for fuzzy diagnosis systems in LTE networks, Expert Systems with Applications
17. Lei JZ, Ghorbani A (2004) Network intrusion detection using improved competitive learning neural network, Proc Second Ann Conf Commun Netw Serv Res
18. Mohammada MN, Sulaimana N, Muhsinb OA (2011) A novel intrusion detection system by using intelligent data Mining in Weka Environment. Proced Comp Sci 3:1237–1242
19. Nguyen HA, Choi D (2008) Application of data mining to network intrusion detection: classifier selection model, APNOMS 2008, LNCS 5297, Springer-Verlag Berlin Heidelberg 2008. pp.399–408
20. Olusola AA, Oladele AS, Abosede DO (2010) Analysis of KDD '99 Intrusion Detection Dataset for Selection of Relevance Features, Proceedings of the World Congress on Engineering and Computer Science Vol. I WCECS 2010. October 20-22, 2010, San Francisco, USA
21. Revathi DAMS (2013) A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection. Int J Eng Res Technol 2(12):1848–1853
22. Roy PK, Paul C, Sultana S (2014) Oppositional teaching learning based optimization approach for combined heat and power dispatch. Int J Electr Power Energy Syst 57:392–403
23. Scarfone K, Mell (2007) Guide to Intrusion Detection and Prevention System, National Institute of Standards and Technology, Special Publication 800–94
24. Sharma N, Mukherjee S (2012) Layered Approach for Intrusion Detection Using NaïveBayes Classifier. ICACCI'12, August 3-5
25. Soon tee T, Ma K-L, Wu SF (2004) Detecting Flaws and Intruder with Visual Data Analysis, Published by the IEEE Computer Society, IEEE Computer Graphics and Applications
26. Sree Kala T, Christy A (2016) "A pattern matching algorithm for reducing false positive in signature based intrusion detection system". Int J Eng Technol 8(2):580–580
27. Sree Kala T, Christy A (2017) "A survey and analysis of machine learning algorithms for intrusion detection system". Journal of Advanced Research in Dynamical & Control Systems 04(Special Issue):40–46
28. Sree Kala T, Christy A (2019) "An intrusion detection system using opposition based particle swarm optimization algorithm and PNN". IEEE EXPLORE, pp 184–88

29. Tiwari KK, Tiwari S, Yadav S (2011) Intrusion Detection Using Data Mining Techniques. Int J Adv Comp Technol (IJACT)
30. Wang BX, Zhang DH, Wang J (2008) Application of Neural Network to Prediction of Plate Finish Cooling Temperature. J Centr South Univ Technol 15(1)
31. Witten IH, Franck E (2005) Data mining practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco
32. Wu D, Pigou L, Kindermans P-J, Le ND-H, Shao L, Dambre J, Odobez J-M (2016) Deep dynamic neural networks for multimodal gesture segmentation and recognition. IEEE Trans Pattern Anal Mach Intell 38(8): 1583–1597
33. Xu X (2006) Adaptive Intrusion Detection Based on Machine Learning: feature Extraction, Classifier Construction and Sequential Pattern Prediction. Int J Web Serv Prac 2(1):49–58
34. Youssef A, Emam A (2011) Network Intrusion Detection Using Data Mining AND Network Behaviour Analysis. Int J Comp Sci Info Technol (IJCSIT) 3(6)

T. Sree Kala obtained her B.Sc degree in Computer Science from the N.M.C.College, Manonmanium Sundaranar University, Tirunelveli, in the year 2002. She received her M.Sc degree in Computer Science from the St.Joseph's College, Bharathidhasan University, Tiruchirappalli, in the year 2005. She received her Ph.D degree in Computer Science from Bharathiar University, Coimbatore at 2020. She worked as a Testing Engineer at OnSpec Electronics Inc, from 2005 to 2008. Presently, she is working as an Assistant Professor in the Department of Computer Science at VISTAS, Chennai. Her specialization is Network Security and Intrusion Detection Systems. She has published many papers in Scopus indexed journals and international journals.

A. Christy completed her Bachelor Degree in Physics from Holy Cross College, Nagercoil and Master of Computer Applications in the year 1990 from Annamalai University, Tamil Nadu which is one of the well-known Universities in the country. She joined as a Lecturer in the Department of Computer Science, Annai Velankanni College in the year 1992 and worked as the Principal of St. Mary's School of Management Studies from 2009 to 2014. Presently, she is working as a professor in Sathyabama Institute of Science and Technology. She has got more than 25 years of teaching experience. She has published many papers in International and National Journals with high impact factor.She has also participated and presented papers in International and National Conferences. Her area of interests includes Text Mining and Web Mining.