Bulletin of Environment, Pharmacology and Life Sciences
Bull. Env. Pharmacol. Life Sci., Special Issue [1]2022 : 915-923
©2022 Academy for Environment and Life Sciences, India
Online ISSN 2277-1808
Journal's URL:http://www.bepls.com
CODEN: BEPLAD

ORIGINAL ARTICLE **OPEN ACCESS**

Double Particle Swarm Optimization based Ensemble ML Technique for detecting the intrusion in Networks

Pallavi S Deshpande*,Satish R Jondhale,M.D.Jakhete, Sarika A Panwar
Department of Electronics and Telecommunication Engineering, Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune, India, PIN: 411043,
Email: pallavisd@rediffmail.com

ABSTRACT

Nowadays, in communication fields, people shares more number of data over internet, which results in increasing size of network with corresponding data. This will automatically leads to security issues by injecting many attacks to the data and provides a huge challenge for the network security for accurately detecting the intrusions. Therefore, researchers introduced the Intrusion Detection System (IDS) as a powerful tool to prevent the attacks using network traffic and preserves the data integrity, confidentiality and availability. However, IDS has problems in detecting the intrusions and improves the false alarm rate, while ensuring the detection accuracy. Therefore, a potential solution is deployed by implementing the machine learning (ML) to solve the issues of IDS. In this research work, an effective Network based IDS (NIDS) is developed by proposing an ensemble ML classifier with feature selection technique. Here, selection of features plays a major role for improving the detection accuracy, where this process is carried out by Double Particle Swarm Optimization (DPSO). The proposed DPSO uses two fitness function to control relevance and redundancy of the selected features and given as an input to the ensemble ML classifiers. The research work uses the NSL-KDD dataset as input for detecting the attacks by using proposed methodology. The simulation results are conducted to test the performance of proposed model with existing PSO model in terms of various parameter metrics. The proposed DPSO achieved 98.30% of accuracy, where PSO achieved 92.05% of accuracy and this is due to the usage of two fitness functions in DPSO.

Keywords: Machine Learning; Double Particle Swarm Optimization; Intrusion Detection System; NSL-KDD dataset; Attacks; False Alarm Rate.

Received 21.02.2022 Revised 23.03.2022 Accepted 12.04.2022

INTRODUCTION

The stability and safety of various systems can be easily affected by Internet nowadays. Researchers finds some static defence mechanisms to provide security to those systems by using firewalls and software updates. In terms of dynamic solutions, IDS is employed by the researchers [1]. The network traffic is continuously observed by IDS for warnings and frightened action, when the explosion of such actions are occurred in the system [2]. Monitors, detects and evaluates malicious events in a computer system or local domain. It includes various risk management options from threats and incidents [3]. The network traffic is monitored and evaluated by NIDS, which is used to access important system files and logs of the most important servers [4]. The risks of network security over the last few years have modified into systematized, more complex and problematic to identify. Moreover, failures to prevent attacks are increasing, which is a violation of the CIA's network security policy of confidentiality, integrity, and accessibility. According to a study by McAfee Labs, in the first quarter of 2019, ransom ware attacks increased by 118%, new families of ransom ware were identified, and threat actors used innovative methods. The targeted attacks of affected servers are dealt with desired instrument called PowerShell and this process is increased by 460%. To deal with this problem, it is significant to have a better IDS in terms of diagnosis speed and accuracy [6]. Vulnerabilities and Threats are managed by the system called IDS [7]. In order to destroy system of any organization or government, group of people or individuals can make a threats. Hacking can produce a product failure, damage to sensitive information, and theft of personal information, money, or other assets [8]. For the advanced threat attacks, NIDS is used to provide a protection, which is considered as key strategy, however, it faces more number of issues. For many years, traditional IDS have been used, the primary feature of which is identification [9-11]. All kind of attacks such as novel varieties are unable to identify by supervised IDS, since it has some restrictions by database's scale rate of predefined labels. In order to solve these issues, researchers introduced ML strategies to develop an efficient IDS model. Semi-supervised learning is a unique form of ML [12]. In a

routine supervised study, several examples are needed to know the exact classifier. However, collection of labelled data is not an easy process [13]. It needs significant time and effort on the part of competent commentators. Getting unlabelled data is relatively easy, but it's hard to use. Both classified and unclassified data were considered in the semi-supervised study, which significantly supports the study's performance [14-15]. By using the labelled and unlabelled data, classification precision can be enhanced in the supervised and semi-supervised learning, due to the innate nature of the data. In this research study, an effective NIDS system is developed using meta-heuristic based feature selection with ensemble ML. A double PSO is used for selecting the optimal features and minimized the irrelevant features from the input data. An ensemble learning of RF techniques are employed as classification technique. The experiments are conducted on NSL-KDD dataset in terms of various parameters for testing the proposed model. The remaining paper is constructed based on the study of existing works that are presented in Section 2. The brief explanation of each steps of proposed model is provided in Section 3, where the validation process takes places in Section 4. Finally, conclusion of the work is given in Section 5 along with future work. According to feature selection and ensemble classifier, an intelligent IDS is developed by Zhou et al. [16]. In order to diagnosis the multi-attacks in the work, correlations are used for feature selection and then ensemble classifiers are used that includes RF, C4.5, Average of Probabilities as AOP rule in Forest by Penalizing Attributes as Forest PA. In experiments, NSL KDD ranks the CICIDS 2017 database and provides an accuracy of 99%. The main drawback of this system is that the author does not evaluate the model in terms of time efficiency. In Karatas et al. [17], a model using SMOTE sampling technique to balance curved categories has been proposed in the CIC-IDS2018 database. The samples of the curved categories increase in proportion to the average sample size. Various ML techniques such as RF, KNN, gradient boosting, adaboost, linear discriminant analysis and decision tree achieved 99% of accuracy, when implemented with this SMOTE technique. In IDS model, feature selection technique called genetic algorithm is used for choosing the important features. The results showed that the models achieved an accuracy of 99%, but he has not evaluated the model recommended for time-based measurements.

Several in-depth study approaches have been proposed to develop effective IDS model. In order to train the neural network, Long Short-Term Memory with Attention Mechanism technique is developed as dynamic anomaly detection system in Lin et al. [18]. CIC-IDS 2018 dataset is used for validation process. From the simulation results, the accuracy rate of 96.2% is achieved by this model, however, time efficiency is not considered in this model.

The Botnet attack is detected by developing an artificial neural network with the help of CIC-IDS 2018 dataset in Kanimozhi V, Prem Jacob [19]. The validation analysis shows that the model achieved 99.97% of high accuracy, 0.999 of area under curve, but this high performance is achieved only on the detection of botnet. Time efficiency is high in this model, which is the major drawback.

The comparison of deep learning (DL) model is studied by Ferrag et al. [20]. The latest CIC-IDS 2018 dataset is used, where the DL techniques includes auto encoders, Boltzmann Machine, Restricted Boltzmann Machine, convolutional neural network, Deep Belief Networks, deep neural network and finally Recurrent Neural Network are considered. Only 5% of the complete database was examined. Concerns about inequality have not been addressed by any approach. In addition, only in-depth study models were evaluated for recall rate and accuracy. No additional measurements such as precision ratio and F1-measure is considered. A fast learning model based PSO as PSO-FLN is developed by Ali et al. [21]. KDD99 dataset is used as input in this research work, where the simulation process proves PSO-FLN achieved better performance than different learning techniques. But, the recommended approach did not detect all types of attacks and did not evaluate the time performance of the model.

Data set considered in this Work

A significant impact is created on KDD99 data set by analyzing various system performance and identified it has two issues. The major drawback is the significant number of recurring records [22], because the learning process is biased towards repetitive entries. This does not allow networks like U2R and R2L to study very small logs, which are usually very harmful. In addition, since these normal records are in the database, the results of the analysis will be based on the techniques with the highest detection rates in the frequently repeated records. Therefore, new database called NSL-KDD is provided by Tavalli et al. [23]. This includes selected entries from the entire KDD database, but the harm is not discussed. Five different categories of data such as normal, Denial of Service as DoS, Probe, User-to-root as U2R and remote to local as R2L [24] are considered in this model, where these four attacks are described as follows:

Inquiry attack: With the intent of bypassing security restrictions, an attacker seeks to collect data on computer networks. An example of inquiry attack is port scan.

DoS attack: The authorized information cannot be able to access by genuine users due to the presence of attacker. An example of this attack is SYN flood attack. A requests of redundant packets are sent by the

attackers to the system to delete its resources, so that system become unable to handle legitimate requests.U2R attack: Initially, an access is gained by the attacker as regular user account, then they took advantage of the system and exposure the root access. Example of this attack is the Xterm exploitation.
R2L attack: Without having an account on particular machine, a packets can be sent to that machine by the attacker, which is happened in this attack. In order to gain remote local access as a user of this device, the attacker exploits some damage. The F tp_write exploit is a type of R2L attack.
An analysis by Revathi [25] showed that the comparison of various IDS methods considered NSL-KDD database as suitable dataset. Therefore, in this paper, the database was selected as research material.

Proposed System

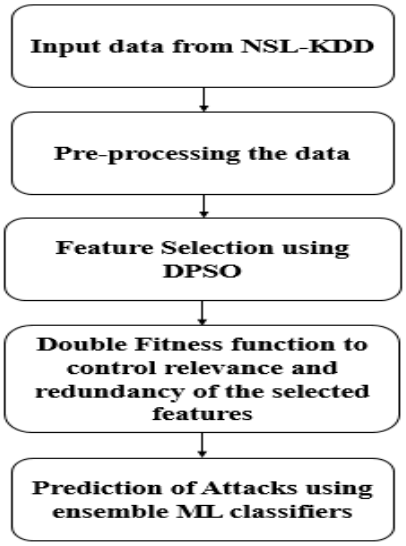

Figure 1: Proposed Methodology's Workflow

Data pre-processing
At this initial stage, the database has been preprocessed. Data wrestling was performed in a complete database to prepare the data for further calculations. The database was later renamed into two categories: attack and non-attack. Zero values have been removed from the database and the number has been reduced to 16.1 million from 16.2 million. Here, some columns are excluded from the database, where the columns includes source port, time stamp, flow ID and IP address. The time of the attack is recorded by timestamp, where source and destination's device IP addresses are recorded by the column IP address. Both columns will be omitted, because trained models should not be biased. The port number of the attacking source device is presented in the column of source port.
A total of 122 data features have been added after the change. The value of num_outbound_cmds data is zero, therefore it is removed after the training dataset analysis. In the original dataset, different feature fields has major variations that affects training outcome. Therefore, Standard Scaler approach is used to standardize the data. The mean is subtracted and divided by the variance for converting the standardized data. The data that has 0 is mean and the value of 1 represents the standard deviation.
Selection of important feature
In the hope that the 2D database would provide a better representation of the database, the dimensions of the NSL-KDD were reduced from 42 to 2 for a conscious understanding of the data distribution. To identify the most important components of the data, principal component analysis as PCA is used by existing techniques, which is a new way to reduce and understand the overall database and reduce information corruption. But, this research work uses the DPSO as feature selection technique, which is explained as follows:
PSO [26]
In order to obtain the DPSO, initially, the basic principle of PSO and binary PSO must be studied, where DPSO defines two different models using fitness function for identifying the relevant features. The basic PSO can be referred from the [26] and here, the Eq. (1-2) shows the updating process for velocity and position vectors to the next iteration t + 1.

$$V_{t+1}^{i} = W \times V_{t}^{i} + C_1 \times r_1(t) \times \left(P_{best}^{i} - P_{t}^{i}\right) + C_2 \times r_2(t) \times \left(G_{best} - P_{t}^{i}\right) \quad (1)$$

$$P_{t+1}^{i} = P_{t}^{i} + V_{t+1}^{i} \quad (2)$$

W is the depression weight constant, which does not allow the effect of the particle velocity on the next iteration to move the particle out of the search space during the next iteration. The constant W is usually in the range [0.4,0.9]. C1 and C2 are called constants and acceleration coefficients. The constants C1 and C2 usually fall in the range [1,5]. At the same time, r1 and r2 are evenly distributed with random values [0,1]. C1, C2, r1 and target for use The r2 constants are used to measure scientific knowledge and social knowledge in velocity changes. Accordingly, all particles can reach the optimal solution to the problem.

Binary PSO

Traditional PSO works well with cascading domains, but can have a detrimental effect on results when managing a unique space. Therefore, Kennedy and Eberhard introduced the binary PSO algorithm as BPSO [27] to deal with this problem.

There are two different limitations are presented in the traditional BPSO algorithm: According to the velocity vector, particles' position is depends in the next iteration. Therefore, a new method is needed to estimate the state of a new particle, taking into account the effect of the current state of the particle. BPSO has a great opportunity for advanced integration while retaining the second joint diversity. Therefore, it is necessary to change the acceleration equation to allow the particle to continue moving towards a better solution. As a result, Zhou et al. [28] proposed binary particle swamp optimization for fitness ratio selection (FPSBPSO) has been to address each of these shortcomings. FPSBPSO will update the particle velocity and position in the next iteration according to equations (3) and (4) [29].

$$V_{t+1}^{i} = \begin{cases} mr, & \text{if } n_0 = 0 \\ 1 - mr, & \text{if } n_1 = 0 \\ \frac{n_1}{n_0 + n_1}, & otherwise \end{cases} \quad (3)$$

$$P_{t+1}^{i} = \begin{cases} 1, & \text{if rand}\,() < V_{t+1}^{i} \\ 0, & otherwise \end{cases} \quad (4)$$

If mr is an independent parameter of the algorithm, n_0 is the current position of the particles, the number of bits associated with the optimal and global individual vectors, and n1 is the inverse of n0, which can be calculated using (3-n 0). Instead of fixing the shortcomings of BPSO, the FPS-BPSO algorithm improved the result of FPSBPSO optimization problems, especially regarding the feature selection process [28]. Also, tuning FPSBPSO is easier than tuning BPSO because it has only one parameter. They concluded that in most cases, a value of 0.01 for each mr parameter would be a good choice.

Using BPSO and information theory, they developed methods for selecting features based on individual target candidates. The first is to assess the suitability and frequency of the subcommittee of the selected features by measuring the mutual information of each pair of features. The first method of exercise can be obtained using equation (5) [29].

$$\text{Fitness}_1 = \alpha_1 \times D_1 - (1 - \alpha_1) \times R_1 \quad (5)$$

$$D_1 = \sum_{x \in X, c \in C} I(x;c),$$

$$R_1 = \sum_{x_i x_j \in X} I(x_i; x_j)$$

Set of defined features is denoted as X and class labels set is represented as C. By defining the interaction between each feature and the class labels, the D_1 subset of the given feature is computed. On the other hand, R_1 determines the mutual information that is shared by each pair of selected features and evaluates the frequency of the subset of features. The purpose of using Fitness1 is to identify subsets of features that are most relevant to the class designations and at the same time have the least frequency with each other. On the other hand, the second method determines the relevance and frequency of the selected subset of features by measuring the entropy of each feature set. The second method can be obtained using equation (6) [29].

$$\text{Fitness}_2 = \alpha_2 \times D_2 - (1 - \alpha_2) \times R_2 \quad (6)$$

$$D_2 = \sum_{c \in C} I\,G(c|X),$$

$$R_2 = \frac{1}{|S|}\sum_{x \in X} IG(x|\{X/x\})$$

X and C are defined as specified in equation (5). According to the entropy, the relationship between selected features and class labels is indicated by D_2. The redundancy R_2 in the selected feature subset is calculated by measuring the combined entropy of all selected features. Fitness 2 is an enhanced fitness activity that reduces repetitions (R_2) and increases fitness at the same time (D_2). In addition, weights 1 and 2 are constant values up to [0,1]. In order to improve the performance of ensemble ML classifiers, two fitness function is used to control the redundancy and relevance of the selected features of NSL-KDD data. The correct value of these parameter is considered as either 0.8 or 0.9, which is proved by experimental results.

Classification using Ensemble ML Classifiers

Random Forest: Freeman [30] randomly presented a forest taxonomist with several taxonomic trees. The class and root tip range are completely consistent with the information acquisition of the attribute segment. The information gain (IG) for dividing a training data set (Y) into subsets (Yi) can be defined as:

$$IG = -\sum_i \frac{|Yi|}{Y} E(Yi) \quad (7)$$

The operator |·| is the scope of the set and E(Yi) is the entropy statistic [37] of the set Yi, defined as:

$$E(Yi) = 1\sum_{j=1}^{N} p_j \log_2(p_j) \quad (8)$$

N is the number of sleep states to be classified (N = 5) and p_j is the ratio (Yi) of sleep state to the group. If the information gain is positive, the node is split. If it is negative, the node will remain the same and it will become the header of the sheet assigned to the class label. Quality is collected with the highest gain in information from the remaining qualities. The separation process will continue until properties are determined. The output of the classification is the most active (active) sleep state in the training nodes extension subgroup [31]. Each node of RF is separated by a positive partition between all features of a given tree [32].

Bagging: merging the results of different methods into one by combining different results. When we talk about rating, the easiest way to implement it is by rating, but if you are dealing with calculating numeric values, that means averaging. Trees can be linked to a vote in each test case. If a class gets more votes than others, it is considered valid. The more votes we get, the more consistent the vote result. When new data sets are added to training, it makes unusually poor decisions and creates trees for them. To create a new database, a random model is used in the instances of the original database. This modeling process avoids and reflects some events. Instead of creating independent databases from the domain, it will republish its own training data [33].

Reinforcement is a technique that involves multiple samples looking at complementary patterns. Vote or increase is used to standardize the results of each sample, like bagging [33].

AdaBoost can learn to handle weighted events, which is a positive number. Event weights are used to calculate the classifier error. When you do this, the learning algorithm will focus on a specific case of overweight. The AdaBoost algorithm starts by giving equal weight to all events of the training data, then prompts the learning system to generate a classification for this data and review each event according to the version of the classifier [33].

MultiBoosting: Here, the combination of Bagging and AdaBoost is used to form this technique. In fact, bagging [34] has a greater effect on variability than adabost, and their combination reduces variability and prevents a reduction in adaptive dependence. Since bagging reduces the number of training examples available to prepare each subgroup, we can use wagging to remove this shortcoming [34]. To set the number of sub-groups, a single-panel, multi-pot size argument is used, which assigns an icon to each member of the final group and specifies the target sub-committee member's token. In addition, MultiBoosting has the advantage of power computation via AdaBoost, where subsets can be studied in parallel, while modification is required in the subset decision pre-learning process. Since AdaBoost is an inherent domain, it reduces the possibility of integration. On the other hand, not all classifiers studied with oscillation depend on others, allowing synchronization, which is the advantage of this algorithm at the subcommittee level.

Rotation Forest: The accuracy of classification can generally be improved by looking at a group of a few classifiers. Rotating Forest is a technique for classifying groups based on the feature extraction process. Initially, the K subset is achieved by dividing the set of feature vectors and PCA is implemented for each subset. Data variance is preserved while all important components are preserved. As a result, new

features of the base classifier are created by rotating the K-axis. Rotation preserves the different supports within the group by extracting the feature for each base classifier. The term 'forest' was chosen as the primary taxon for definitive trees (DDs) because of their sensitivity to alternation. The safety of all major components maintains individual precision. In addition, accuracy is preserved as the entire database is used to train each base classifier. Instead of using the AdaBoost, Random Forest, and Bagging approaches, this groups can develop more accurate and distinct individual classifications [36].

RESULTS AND DISCUSSION

In this section we discuss about the simulation experiments are performed by using the tool of python 3.7.3. The hardware such as PC with capacity of core i5 processer and storage range of 8 GB RAM. These things are used to validate the performance parameter such as accuracy, precision, recall and F-measure. However, these parameters are analyzed by using NSL-KDD dataset.

Performance Metrics

The simulation analysis of recommended scheme and performance metrics outcomes are given in following study. In each parameter outcome is calculated by using formula. These are followed in the fellow section. The precision and recall parameter formulation is derived from the following equation of (9) and (10).

$$\text{Precision (PR)} = \frac{|\text{Data correctly classified to the class c}|}{|\text{Total data classified to class c}|} \quad (9)$$

$$\text{Recall (R)} = \frac{|\text{Data correctly classified to the class}|}{|\text{Total Data in class c}|} \quad (10)$$

Accuracy is the standard deviation of statistics and description of random errors. The overall accuracy of the text classification results for determining infiltration is given in the equation (11).

$$\text{Accuracy} = \frac{|\text{Total correctly classified Data}|}{|\text{Total number of Data}|} \quad (11)$$

F Measure is a precision test measure and takes into account both accuracy and test recall for calculating a score. The overall F- measure formula is derived in the equivalence (12).

$$F - \text{Measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (12)$$

Performances Evaluation of proposed model

Table 1 and Figure 2 shows the performance of proposed ensemble ML techniques with feature selection techniques in terms of PR, R, ACC, F-M and False Positive Rate (FPR), Figure 3 shows the graphical representation of proposed model in terms of FPR.

Table.1. Proposed Classification System evaluation with Double PSO.

Techniques	PR	R	F-M	FPR	ACC
Random Forest	97.32	97.54	97.42	11.35	97.55
Bagging	97.86	97.64	97.35	11.60	97.60
AdaBoost	97.90	97.89	97.87	08.30	98.01
MultiBoosting	97.12	97.64	98.01	08.20	98.05
Rotation Forest	98.05	98.25	98.23	08.04	98.30

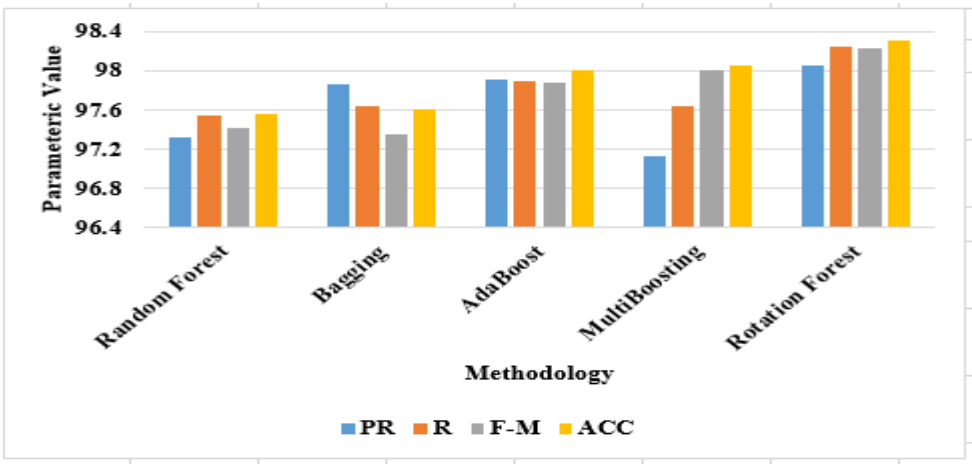

Figure 2: Graphical Representation of ensemble ML with DPSO

Among the various ensemble techniques, Rotation forest achieved better performance in terms of PR and R, for instance, it achieved 98.05% of PR and 98.25% of R and other techniques achieved nearly 97% of PR and R. The reason is that rotation algorithm provides more accurate results on continuous features and uses the K subset to identify and preserve the information variability. The multiboosting technique achieved 98.23% of F-M, where RF, bagging and adaboost classifiers achieved nearly 97% of F-M and Rotation forest achieved 98.23% of F-M, which is higher than other ensemble ML techniques. Here, DPSO is used for feature selection with two fitness function for effective removal of redundancy features in the dataset. Moreover, due to the advantages of rotation forest over other algorithms and it is combined with DPSO, it produced more accurate results than other techniques. Finally, in the analysis of accuracy experiments, RF and bagging achieved nearly 97.50%, Adaboost achieved 98.01%, multiboosting achieved 98.05% and rotation forest achieved 98.30% of accuracy. From these analysis, it is clearly proves that Rotation forest achieved better performance, where RF achieved less performance than other technique and this is due to more computation power are required to build the trees to combine the outputs and took more time for training to determine the class.

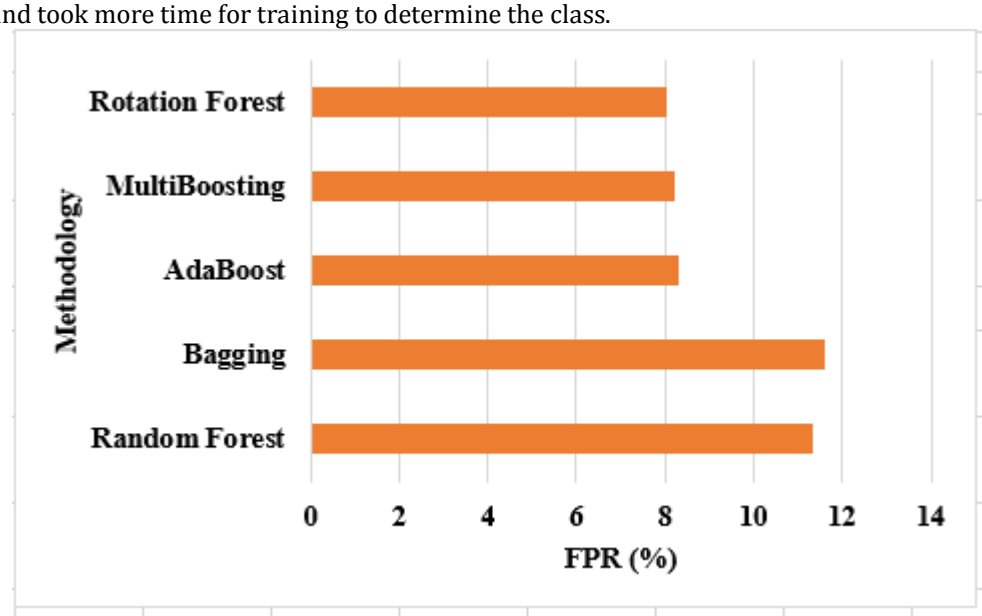

Figure 3: Graphical Representation of ensemble classifiers in terms of FPR

Initially, the RF and bagging has highest FPR, i.e. they achieved nearly 11%, where Adaboost and multiboosting achieved 8.25% of FPR. When comparing with these ensemble classifiers, rotation forest achieved only 8.04% of FPR and effectively classifies the data into normal or attacks. Therefore, Table 2 and Figure 4 shows the comparative analysis of proposed feature selection techniques with other techniques in terms of FPR. Here, rotation forest is considered and tested with proposed DPSO and other algorithms, because it only achieved better performance than other ensemble ML classifiers.

Table.2. comparative analysis of various optimization techniques with Rotation Forest.

Techniques	PR	R	F-M	FPR	ACC
GA- Rotation Forest	91.45	90.89	91.90	10.08	92.10
PSO-Rotation Forest	90.80	91.80	91.70	10.50	92.05
Bacterial Foraging Optimization (BFO)-Rotation Forest	91.70	91.90	91.95	10.00	92.20
Artificial Bee Colony (ABC)-Rotation Forest	90.25	90.21	90.07	10.05	90.80
Ant Colony Optimization (ACO)-Rotation Forest	90.21	90.28	90.75	10.35	90.65
Gravitational Search Algorithm (GSA)-Rotation Forest	90.45	90.43	90.90	9.60	91.23
Mayfly Optimization Algorithm (MFO)-Rotation Forest	93.05	93.23	94.07	9.30	94.09
Whale optimization (WOA)- Rotation Forest	94.24	94.78	94.90	8.2	94.90
DPSO-Rotation Forest	98.05	98.25	98.23	08.04	98.30

Figure 4: Graphical Representation of Proposed DPSO in terms of FPR

From this various existing algorithms, GA, PSO, BFO, ABC, ACO and GSA achieved nearly 90% of PR, 90.50% of R, 91% of F-M, 91.50% of ACC and 10% of FPR. The reason is that GA is time-consuming and formulation of fitness function along with the use of population size must be carefully chosen, where BFO makes the fixed step size for balancing the exploitation and exploration. However, there are three limitations presents in ACO that includes convergence speed, stagnation phase and rate of exploration and exploitation is high. Due to this drawbacks, rotation forest algorithm also provides low performance, when it is implemented with other algorithms. In the PR and R analysis, MFO, WOA and DPSO achieved 93.05%, 94.24% and 98%, where FPR of MFO, WOA and DPSO achieved 9.30%, 8.2% and 8.04%. Both MFO and WOA achieved nearly 94% of accuracy, where DPSO achieved 98% of accuracy and this is due to the usage of double fitness model to avoid the early convergence. From these experiments, it is clearly proves that proposed model achieved better performance in terms of all parameters with existing techniques.

CONCLUSION

In this research work, an effective NIDS is developed by using feature selection technique with ensemble ML classifiers. Initially, NSL-KDD dataset is used as input and pre-processing is carried out to remove the missing data and normalize the data by using mean-standard deviation process. The features are selected and irrelevant data are removed by using DPSO, which uses two fitness function. Finally, those input given to ensemble classifiers for predicting whether the data is normal or attacked. The simulation is done by using Python software and uses five different metrics to check the efficiency of proposed DPSO with ensemble ML classifies. From the analysis, it is clearly proves that rotation algorithm with DPSO achieved better performance than existing techniques and other classifiers. However, the research work focused only detecting normal and attacks data, where more number of attacks are occurred in software such as black-hole, botnet, jelly fish attacks, rushing attacks, etc. Therefore, the model is improved with deep learning classifiers to handle these kinds of attacks and provides the prevention mechanism.

REFERENCES

1. Hodo, E.; Bellekens, X.; Hamilton, A.; Dubouilh, P.L.; Iorkyase, E.; Tachtatzis, C.; Atkinson, R.: (2016). Threat analysis of Iot networks using artificial neural network intrusion detection system. In: 2016 International Symposium on Networks, Computers and Communications (ISNCC) pp. 1–6.
2. Sharma, P.; Sengupta, J.; Suri, P.: (2019). Survey of intrusion detection techniques and architectures in cloud computing. Int. J. High Perform. Comput. Netw. 13(2), 184.
3. Modi, C.; Patel, D.; Borisaniya, B.; Patel, H.; Patel, A.; Rajarajan, M.:(2013). A survey of intrusion detection techniques in cloud. J. Netw. Comput. Appl. 36(1), 42.
4. Salah, K.; Calero, J.M.A.; Zeadally, S.; Al-Mulla, S.; Alzaabi, M. (2012). Using cloud computing to implement a security overlay network. IEEE Secur. Priv. 11(1), 44.
5. Kamarudin, M.H.; Maple, C.; Watson, T.: (2019). Hybrid feature selection technique for intrusion detection system. Int. J. High Perform. Comput. Netw. 13(2), 232.
6. HaddadPajouh, H.; Dehghantanha, A.; Khayami, R.; Choo, K.K.R.: (2018). A deep recurrent neural network based approach for internet of things malware threat hunting. Future Gener. Comput. Syst. 85(1), 88.
7. Zouhair, C.; Abghour, N.; Moussaid, K.; El Omri, A.; Rida, M.: (2018). A review of intrusion detection systems in cloud computing. In: Security and Privacy in Smart Sensor Networks,pp. 253–283.

8. Diro, A.A.; Chilamkurti, N.: (2018). Distributed attack detection scheme using deep learning approach for internet of things. Future Gener. Comput. Syst. 82(1), 761.
9. Kasongo, S.M.; Sun, Y.: (2020). A deep learning method with wrapper based feature extraction for wireless intrusion detection system. Comput. Secur. 92(1), 101752.
10. Park, S.T.; Li, G.; Hong, J.C.: (2018). A study on smart factory-based ambient intelligence context-aware intrusion detection system using machine learning. J. Ambient Intell. Humaniz. Comput. 11(4), 1:90-99..
11. Roman, R.; Lopez, J.; Mambo, M.: (2018). Mobile edge computing, fog et al.: a survey and analysis of security threats and challenges. Future Gener. Comput. Syst. 78, 680.
12. Goldstein, M.; Uchida, S.: (2016). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. PLoS ONE 11(4), e0152173.
13. Stojanovi´c, B.; Hofer-Schmitz, K.; Kleb, U.: (2020). Apt datasets and attack modeling for automated detection methods: a review. Comput. Secur. 92(1), 101734.
14. Elakkiya, E.; Selvakumar, S.: Gamefest: (2019). Genetic algorithmic multi evaluation measure based feature selection technique for social network spam detection. Multimed. Tools Appl. 97, 1 :12-17.
15. Bostani, H.; Sheikhan, M.: (2017). Hybrid of anomaly-based and specification-based ids for internet of things using unsupervised based on map reduce approach. Comput. Commun. 98, 52..
16. Zhou Y, Cheng G, Jiang S, Dai M. (2020). Building an efficient intrusion detection system based on feature selection and ensemble classifer. Comput Netw. https://doi.org/10.1016/j.comnet.2020.107247.
17. Karatas G, Demir O, Sahingoz OK. (2020).Increasing the performance of machine learning-based IDSs on an imbalanced and up-to-date dataset. IEEE Access. ;8:32150–62.
18. Lin P, Ye K, Xu C-Z. (2019). Dynamic network anomaly detection system by using deep learning techniques. Cloud Comput CLOUD 2019. https://doi.org/10.1007/978-3-030-23502-4_12.
19. Kanimozhi V, Prem Jacob T. (2109). Artificial Intelligence based Network Intrusion Detection with hyper-parameter optimization tuning on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing. ICT Express.;5(3):211–4
20. Ferrag MA, Maglaras L, Moschoyiannis S, Janicke H. (2020). Deep learning for cyber security intrusion detection: approaches, datasets, and comparative study. J Inform Security Appl. 50:102419.
21. Ali MH, al Mohammed, B. A. D., Ismail, A., & Zolkipli, M. F. (2018). A new intrusion detection system based on fast learning network and particle swarm optimization. IEEE Access. 6:20255–61.
22. Cup, K.: (2007).http://kdd.ics.uci.edu/databases/kddcup99/kddcup99. html.
23. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A.: (2009). A detailed analysis of the kdd cup 99 data set. In: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, pp. 1–6.
24. Ravi, N.; Shalinie, S.M.: (2020). Learning-driven detection and mitigation of ddos attack in iot via sdn-cloud architecture. IEEE Internet Things J. 7(4), 3559.
25. Revathi, S.; Malathi, A.: (2013). A detailed analysis on nsl-kdd dataset using various machine learning techniques for intrusion detection. Int. J. Eng. Res. Technol. (IJERT) 2(12), 1848.
26. R. Eberhart, J. Kennedy,(1995). A new optimizer using particle swarm theory, in: MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, IEEE, pp. 39–43.
27. J. Kenedy, R. Eberhart, (1997). A discrete binary version of the particle swarm optimization, Computational cybernatics and simulation 5 (1), 4104–4108.
28. Z. Zhou, X. Liu, P. Li, L. Shang, (2014). Feature selection method with proportionate fitness based binary particle swarm optimization, in: Asia-Pacific Conference on Simulated Evolution and Learning, Springer, pp. 582–592.
29. L. Cervante, B. Xue, M. Zhang, L. Shang, (2012). Binary particle swarm optimisation for feature selection: A filter based approach, in: 2012 IEEE Congress on Evolutionary Computation, IEEE, pp. 1–8.
30. Breiman, L. (2001). Random forest. Mach. Learn.45, 5–32.
31. Fraiwana, L.; Lweesyb, K.; Khasawnehc, N.; Wendz, H.; Dickhause, H. (2012). Automated sleep stage identification system based on time-frequency of single EEG channel and random forest classifier. Comput. Methods Progr. Biomed. 108, 10–19.
32. Kalmegh, S. (2015). Analysis of WEKA data mining algorithm REPTree, Simple CART and RandomTree for classification of Indian news. Int. J. Innov. Sci. Eng. Technol. 2, 438–460.
33. Hall, M.; Witten, I.; Frank, E. (2011). Data Mining: Practical Machine Learning Tools and Techniques; Kaufmann: Burlington, NJ, USA.
34. Bauer, E.; Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Mach. Learn. 36, 105–139.
35. Webb, G.I. (2000). Multiboosting: A technique for combining boosting and wagging. Mach. Learn. 40, 159–196.
36. Rodríguez, J.; Kuncheva, L.; Alonso, C. (2006). Rotation forest: A new classifier ensemble method. IEEE Trans. Pattern Anal. Mach. Intell. 28, 1619–1630.

CITATION OF THIS ARTICLE

Pallavi S Deshpande,Satish R Jondhale,M.D.Jakhete, Sarika A Panwar. Double Particle Swarm Optimization based Ensemble ML Technique for detecting the intrusion in Networks Bull. Env.Pharmacol. Life Sci., Spl Issue [1] 2022 : 915-923