IMPLEMENTATION OF SWARM OPTIMIZED ARTIFICIAL NEURAL NETWORK FOR NETWORK INTRUDER DETECTION AND ATTACK CLASSIFICATION

K.Kanaka Vardhini
Assistant professor, Department of CSE,
DRK college of engineering and technology, Hyderabad, india
kanakavardhini@gmail.com

Dr. T.Sita Mahalakshmi
Professor, Dept of CSE, GITAM University, Visakhapatnam

Abstract - In the current era of technological advancement, the usage of computers and internet has increased drastically in the everyday life. With many crucial services operating online, the risk of cyber-attacks has made the use of online security measures inevitable. Identification of the type of the attack is an important task in safeguarding the system. Thus researchers started developing a solution for this problem and an Intruder Detection System (IDS) was created. IDS identify the suspicious behaviour in the network and improve the integrity, reliability and robustness of the host system. The IDS inspect the parameters of the system to detect any intrusion. This paper proposes an IDS using swarm optimization enhanced Artificial Neural Network (ANN) for improved performance. In this paper, KDDCUP dataset and a free dataset from kagge.com have been used for experimentation. First dimensionality reduction is applied using Principal Component Analysis (PCA). This data is fed to the Swarm Optimized ANN (SOANN) for classification. The ANN uses grey wolf optimization algorithm to optimize the weights over multiple iterations. The accuracy of the system is considered as the cost function. This cost function is implemented as the stopping criteria to terminate the optimization algorithm upon reaching the maximum accuracy and the obtained weights are used to train the final system. The performance of the system is compared against various other algorithms and has yielded better results.

***Keywords*:** Intruder Detection System; Swarm Optimization; Artificial Neural Network; Differential Evolution; Principal Component Analysis.

1. Introduction

In the modern era, technology improvements have led to the faster information access over the internet. The ease of access of the data has also improved by the introduction of mobile computing platforms.The number of e-commerce companies are also increasing in the market.This has led to the increasing use of servers and sensitive information is transferred using the internet. This in turn has led to the increase in the threat from the attackers. Such cases increase the concern of developers and industry to develop measures to prevent such attacks. The attackers are using Virtual Private Networks (VPN) to mask the IP and MAC addresses during an attack on a network and thus keeping their identity hidden. Several types of network frauds have been carried out over the internet. Some of them includes spams, website defacements [1], probing etc. Several other network issues have like R2L (Remote to Local), U2R (User to Root) have been discusses by the authors in [2-3]. A solution to this problem is to use the statistical parameters in the network to detect the intrusions caused.

The suggested model would be controlling the traffic of network by curbing malicious efforts. It would seek for solving the issue of Spam, delay in response of server, illicit access to the network resources etc. This is applicable in many Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) [4-5]. A parameter that could be affecting the process of the IDS is noise. The noise affecting the system could raise false alarms as attacks. In the present day traffic scenario, some of the real attacks may get miss classified under the class of false alarm, increasing the False Alarm Rate (FAR) [6]. IDS system cannot detect attacks in the systems with weak authentication or identification. Even encrypted packets cannot be handled [7].

The two types of the IDS systems are:

- Signature based IDS (S-IDS)
- Anomaly based IDS (A-IDS)

S-IDS is based on predefined parameter sets collected from previous attacks. On the other hand A-IDS analyses the network traffic and interprets the intruder. This is the most prevalent method used in the industry as it is dynamic and can be programmed to learn from the new attacks on a regular basis [9]. The signature based IDS is not capable of detecting severe attacks on the system. IDS would be providing the information about the source of the sender. But fake IP would be received in the case where the packet is encapsulated orVirtual Private Network (VPN) is utilized.

This work would attempt for solving few of the above stated issues. The issue is to identify any malicious type of activity or attack on the network utilizing anomaly dependent statistical methods, & relevant features selection depending on the acquired outcomes. The selection of features would be playing an important role in the elimination of irrelevant and redundant attributes, so that it would be choosing the relevant attributes to build the model. KDD Cup data set and other data sets have been considered for the experimentation.

The paper has been organized into the following sections. Related literature review has been presented in section II. Section III contains the proposed framework. Section IV presents the experimental results followed by conclusion and references.

2. Related work

Horng*et al.,* developed a classifier utilizing hierarchical clustering and SVM [10]. The authors constructed the Clustering Feature (CF) tree using hierarchical Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH). The method used has consumed less time for operation and showed promising results in clustering the data. Amiri*et al.,* deployed the mutual information scheme for determining parameter relation by utilizing the forward algorithm of feature selection [11]. The linear correlation coefficients of the parameters are extracted along with the correlation information between linear and non-linear measurements. Least Squares Support Vector Machine was used to detect the intrusion activity in the system.

Gouveia et al. used the features extracted to train the machine learning algorithms [12]. The research was carried out using many boosting based techniques and several other algorithms. Al-mamory*et al.* exercised dissimilar data mining algorithms [13]. In this research, the author proved that random forest algorithm has provided better results in detecting DoS attacks and Fuzzy logic classifier has produced better results in detecting the Probe attack. Deepik*et al.* Used *K*NN classifier &Dempster–Shefer(DS) Theory along with fuzzy logic [14]. DS evidence theory is based on the probability model of the parameters. The analysis depends on the subjectivist view of the probability utilization.

Subba*et al.* used regression and linear discriminate analysis [15]. LDA was used to reduce the dimension of the input data to improve the accuracy. Davaraju*et al.* used neural network for classifying the data of KDD Cup Dataset[16]. The paper has utilized many varieties of neural networks like Probabilistic, Feed Forward, RadialBasis etc. Olusola*et al.* used the concept of entropy to extract relevant features [17] and the concept of rough set theory for the process of classification of the attack. Chitrakar*et al.* used the concepts of clustering like k means and k medoids and again classified the data using Naive Bayes classifier [18].

3. Method and Framework

The most critical step in constructing of a machine learning model for the purpose of classification is the pruning of features which may lead to the reduction of accuracy. It would be helpful in improving the accuracy through True Positive Rate(TPR), False Rejection Rate (FRR), precision and recall of the classifier. Serveral methods have been used by the researchers to carry out the task of feature selection. Some of the methods used extensively are Mutual Information or Info-Gain, Gain Ratio, PCA, Correlation etc. The proposed framework starts with the processing and normalization of the data collected. The input data is first processed and then sent to reduce the dimensionality using PCA. PCA algorithm is applied on the processed data to reduce the redundancy in the data. Fig 1 shows the block diagram of the proposed technique.

3.1 Pre-processing

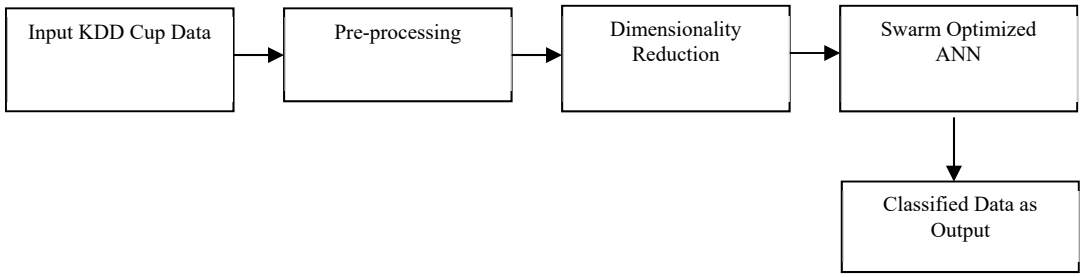

Fig. 1. Proposed framework

In the pre-processing stage, the attributes of protocol type, service and flag are replaced with the corresponding probability values. The numerical operations can't be applied directly on categorical data, data has to be converted into numerical representation. Hence the mentioned three fields are replaced by their equivalent probability value.

3.2 Principal Component Analysis

Principle component analysis or PCA is one of the most important dimensionality reduction techniques. Dimensionality reduction techniques are generally used when the data to be processed is very large. Processing such large data takes a lot of time and computer resources. Dimensionality reduction techniques reduce the size of the data. Dimensionality reduction techniques reduce the number of columns present in the data retaining the important features needed for classification. In any data analytics scenario, the number of dimensions that data has is inversely proportional to the accuracy that is produced at the output. The dimension of the data and the accuracy of classification are inversely related. The accuracy of classification reduces when the features of the data set are too large. Thus dimensionality reduction techniques are used to improve the accuracy of the system.

Step 1: Construct a matrix with all the data vectors stacked beside one another.

Step 2: Calculate the mean feature vector from the matrix.

Step 3: Compute the covariance matrix using the mean vector using equation (1)

$$C_{Matrix} = \sum_{i=1}^{n}(x_i - m) * (x_i - m)^T \quad (1)$$

Where m: mean vector calculated as shown in equation (2)

$$m = \frac{1}{n} * \sum_{1}^{n} x_i \quad (2)$$

Step 4: Calculate the Eigen values and Eigen vectors from the covariance matrix using equation (3)

$$|C_{Matrix} - \lambda I| = 0 \quad (3)$$

The solution to the equations produces eigenvalues$\lambda_1, \lambda_2, \ldots \lambda_n$. Substitute the eigen values in the characteristic results of Eigenvectors.

Step 5: Sort the Eigen values in the descending order and extract the top "*k*" Eigen vectors corresponding to Eigen values.

Step 6: Project the input data on to the *k* vector space to obtain the dimensionality reduced data.

3.3 Feed forward Neural Network

An artificial neural network is a computing model based on biological neural networks. Within the network, neuron groups are combined in several layers and connected to one another. A neural network always has one input layer and one output layer. In between there are theoretically any number of hidden layers. The neurons of neighboring layers are connected to each other. These connections are initially given a random weighting. The weighting indicates the influence of the connected neurons. Hidden layer neurons always have an input and an output. The entire input of a neuron, called the network input, determines a propagation function, for example the linear combination. The linear combination adds up the inputs multiplied by the respective weights. The only exception to this is the input layer.

Weight adjustment takes place with every learning process based on certain learning rules. These adjust the weights of the connections of the neurons so that the output of a model approaches the desired output more and more. A feedforward neural network is an artificial neural network in which the information is passed on in layers from the input to the output layer. It is important that the information is always passed in the direction of the output layer and never in the direction of the input layer.

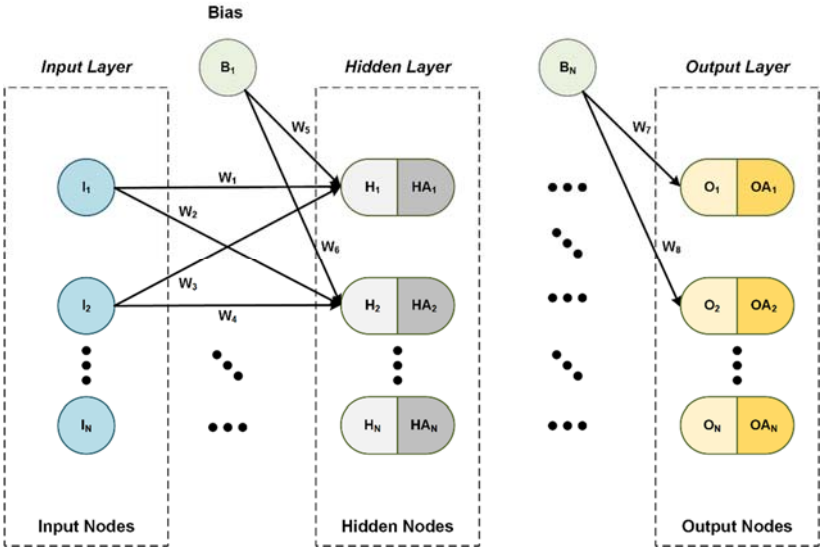

Fig. 2. Proposed framework

Fig 2 shows a typical representation of Feedforward neural network. A neuron present in the first hidden layer has inputs coming from all the previous entities in the input layer. Additionally, each neuron will also have a bias for adjusting the weights. If the number of inputs that a neuron has is *n*, it has *n* weights to be updated. The network takes in all the weights and biases as inputs and finds the output vector. This is called a feedforward process. So each neuron updates the weights over the iterations and finally, the predicted output is obtained. The goal of the network is to reduce the error in the prediction. This is achieved by optimizing the cost function.

3.4 Cost function

The type of classification needed for the proposed problem is multi class classification. In such a problem, the input data needs to be divided into multiple classes. In the training phase, the weights of the neurons in the layers are updated in every iteration. After the updating process, the predicted result is verified with the actual result. The cost function here calculates the accuracy of the predicted values after every iteration. The accuracy is dependent on the number of labels that are classified correctly. In general accuracy is defined in terms of True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). The accuracy is defined as (TP + TN) / (TP + TN + FP + FN). The error value is obtained by subtracting the accuracy value from 1. This is represented in equation (4). The error has to be minimized in an optimal manner by subtracting equation (4) from 1.

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \quad (4)$$

Swarm optimization techniques have been used in the proposed method to optimize the cost function. The algorithms implemented are:

- Genetic Algorithm
- Whale Optimization
- Grey wolf optimization

These optimization techniques are discussed in detailed in the following sections.

3.5 Genetic Algorithm

The genetic algorithm is designed to find out the fittest contestants from a population. In this process, the offspring's are generated from parents through the process of mutation and cross over. These should have better fitness than the parents in order to the move the algorithm further. The process is repeated until the fittest individuals are found.

Five phases are considered in a genetic algorithm.

1. Initial population: The Neural Network(NN) is initialized with random weights for the layers which become the initial population.
2. Fitness function: In the first stage and after every iteration, the cost of the NN is calculated with the weights generated at after each layer. The Fitness function evaluates the weights and thus decides the termination of the program.
3. Selection: In this stage, the fittest weights are selected and passed through further processing.

4. Mutation: In this step, new mutated weights are formed based on the existing ones in hope of achieving the fittest weights.
5. Crossover: The mutated weights are shuffled and flipped in this process.

The steps involved in the process are as follows:

1. Initialize the NN with random weights as initial population.
2. Check the fitness of the weights for creating a reference.
3. Create new population using mutation and crossover.
4. Rank the new population based on the fitness value.
5. Check the termination criteria after each iteration.
6. End the process after the criteria is met.

3.6 Whale Optimization Algorithm (WOA)

The whale optimization algorithm consists of the following main stages

- Encircling prey: The first step in the WOA is the encircling the action that the humpback whales perform on the prey. As the optimal position of the whale cannot be determined in the first instant, the algorithm assumes the present position in the best possible solution.
- Bubble-net attacking method (exploitation phase): Shrinking encircling mechanism and Spiral updating position
- Search for prey (exploration phase).

The algorithm below explains how the whale optimization algorithm is used to optimize the NN weights:

Initialize the data, number of iterations and the number of search agents.

$Initializethewhalespopulation Xi(i = 1, 2, \dots, n) toupdatetheNNweights$

$Initialize a, A, C, landp, where,$

- $a - definesrandomvalueinterval$
- p – probability
- A, C are coefficient vectors

$Calculatethefitnessof each\ searchagent$with respect of the NN wei
$X* = thebestweight$
$while\ (it < Maxiter)$
$foreachsearchagent$
$if\ (p < 0.5)$
$if\ (|A| < 1)$
$Updatetheweightofthecurrentsearchagentbytheequation\ (5)$
$elseif\ (|A| \geq 1)$
$Selectarandomsearchagent\ (X\_rand)$
$Updatetheweightofthecurrentsearchagentbytheequation\ (6)$
end
$elseif\ (p \geq 0.5)$
$Updatetheweightofthecurrentsearchbythebytheequation\ (7)$
end
end
$Calculatethefitnessofeachsearch\ agent$
$UpdateX* ifthereisabettersolution$
$it = it + 1$
$Updatea, A, C, landp$
$endwhile$
$returnX*$

The equations stated in the algorithm are shown in equations (5), (6) and (7):

$$\vec{D} = \left|\vec{C}.\overrightarrow{X_p}(t) - \vec{X}(t)\right| (5)$$

$$\vec{X}(t+1) = \overrightarrow{X_p}(t) - \vec{A}.\vec{D} (6)$$

Where t indicates the current iteration, $\vec{A}$ and $\vec{C}$ are coefficient vectors,

$\overrightarrow{X_p}$is the position vector of the best fitness weight, and $\vec{X}$ indicates the position vector of a whale

The vectors$\vec{A}$ and $\vec{C}$ are calculated as follows:

$\vec{A} = 2\vec{a}.\overrightarrow{r_1} - \vec{a}$

$\vec{C} = 2.\overrightarrow{r_2}$

Where $\vec{a}$ ranges from 2 to 0 and $\overrightarrow{r_1}$ and $\overrightarrow{r_2}$ are random values in [0,1].

$$\vec{X}(t+1) = \overrightarrow{D'}e^{bt}\cos(2\pi t) + \overrightarrow{X^*}(t) \quad (7)$$

Where $\overrightarrow{D'} = \overrightarrow{X^*}(t) - \vec{X}(t)$ indicates the distance between i-th whale and the prey (best solution obtained so far), b is a constant for defining the shape of the logarithmic spiral, and t is a random number in [-1,1].

$$\vec{D} = \vec{C}\overrightarrow{X_{rand}} - \vec{X} \quad (8)$$

$$\vec{X}(t+1) = \overrightarrow{X_{rand}} - \vec{A}\vec{D} \quad (9)$$

Where $\overrightarrow{X_{rand}}$ is a random position vector (a random whale).

3.7 Grey wolf optimization (GWO)

Researchers have based this algorithm based on wolf's hunting pattern. Four categories of wolfs are assigned values over the iterations namely alpha, beta, delta, and omega. The attack process consists of stages like searching, encircling and attacking. The following steps present the detailed step wise mechanism of the GWO.

$Initialize the grey wolf population Xi (i = 1, 2, \ldots, n) with NN weights$
$Initialize a, A, and C, where \vec{A} and \vec{C}$ are coefficient vectors, and $a \in [0,2]$
$Calculate the fitness of each search agent$
$\overrightarrow{X}\_\alpha = the best search agent$
$\overrightarrow{X}\_\beta = the second best search agent$
$\overrightarrow{X}\_\delta = the third best search agent$
$while\ (t < Max number of iterations)$
$for each search agent$
$Update the position of the current search agent$
$end for$
$Update \alpha, A, and C$
$Calculate the fitness of all search agents$
$Update X\_\alpha, X\_\beta, and X\_\delta$
$t = t + 1$
$end\ while$
$return X\_\alpha$

The grey wolf optimizer is used as the proposed method to optimize the NN weights. The following section presents the experiments done on two datasets using this proposed method.

4. Experimental Results

The experiments have been conducted on two datasets. The first dataset is the widely used KDD CUP dataset. The second dataset is downloaded from https://www.kaggle.com/. The details of the datasets and the results are explained below.

Dataset 1:

The KDD Cup dataset consists of 5 types of intrusions. The feature vectors for each intrusion are of length 41. After pre-processing and dimensionality reduction the final data vector is of length 15. The reduced data is set fed to the neural network for training.

Table 1: Accuracy

Dataset	Number of Training Samples	Number of Testing Samples	Training Accuracy	Testing accuracy
KDD	125973	22238	93.28	91.60

Table 1 shows the accuracy of the proposed technique on the KDD Cup dataset. The number of training samples is 125973 and the number of testing samples is 22238. The testing accuracy of the proposed system is 91.60. The detailed class wise accuracies are presented in the form of confusion matrix in table 2.

Table 2: Confusion Matrix

Normal	DOS	R2L	U2R	Probing	Accuracy of each attack
6934	0	0	5	573	6934/7512 = 92.30
0	140	0	0	24	140/164 = 85.37
3	0	2009	1	250	2009/2263 = 88.77
77	0	0	1309	734	1309/2120 = 61.74
130	0	0	69	9980	9980/10179 = 98.04

Total Accuracy: 91.60

Neural Network Analysis

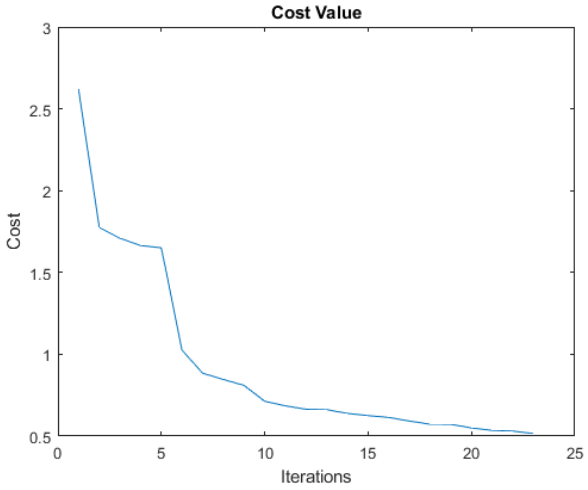

Fig. 3. Cost value pf the NN weight over iteration

- Input size – 15
- Number of classes – 5

Fig 3 shows the cost function value of the proposed algorithm. During the training phase, the cost function of the NN reduced to 0.5 after 20 iterations. This indicates that the weights of the neural network are optimal in classifying the given input data. The comparison of the proposed algorithm with other NN optimizations is listed in table 3.

Table 3: Comparison Results with complete KDD dataset

S. No.	Optimization Algorithm	Train Accuracy	Test Accuracy
1.	Genetic	95.54	93.17
2.	Whale	88.49	82.97
3.	Proposed Grey Wolf	93.28	91.60

Dataset 2

The second dataset is taken from the website Kaggle (https://www.kaggle.com/sampadab17/network-intrusion-detection). The dataset consists of 4 types of intrusions and a normal case. The feature vectors for each intrusion are of length 41. After applying PCA, the length of feature vector is reduced to 15. The data is set fed to the neural network for training. The accuracy of the proposed method over the dataset 2 is presented in table 4.

Table 4: Accuracy

Dataset	Number of Training Samples	Number of Testing Samples	Training Accuracy	Testing accuracy
network-intrusion-detection	125973	10000	94.04	92.08

Table 5 shows the confusion matrix of the proposed algorithm. The accuracy for each category off the attack in the testing dataset is shown here.

Table 5: network-intrusion-detection confusion matrix

Normal	DOS	R2L	U2R	Probing	Accuracy of each attack
3981	51	197		0	3981/4229=94.13
0	3132	200	0	0	3132/3332 = 87.99
0	159	931	13	0	931/1103 = 84.40
0	116	44	1119	0	1119/1279 = 87.49
0	0	10	2	45	45/57 = 78.94

Neural Network Analysis

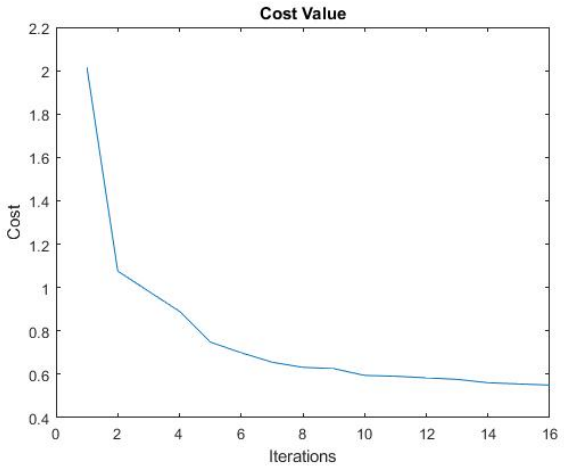

Fig. 4. Cost value of the NN weights over iteration

- Input size – 15
- Number of classes – 5

Fig 4 shows the cost function value of the proposed algorithm. During the training phase, the cost function of the NN reduced to 0.6 after 15 iterations. This lower cost function is an indication towards optimal training of the weights. The comparison of the proposed algorithm with other NN optimizations is listed in table 6.

Table 6: Comparison Results with complete dataset

S. No.	Optimization Algorithm	Train Accuracy	Test Accuracy
1.	Genetic	94.06	92.22
2.	Whale	93.28	90.63
3.	Proposed Grey Wolf	94.05	92.08

The experimental results on both the datasets prove that the proposed method of SOANN produces better result in classification of the attacks. The accuracies are shown in tables 3 and 6.

5. Conclusion

An Intruder Detection System needs to detect anomalies in the network based on the input parameters. This paper proposed a Swarm optimization based NN for the purpose of identifying the attacks. The experiments have been carried out on KDD Cup data set and Kaggle dataset. The proposed method produced better accuracy when compared to the existing optimization techniques.

References

[1] Hoang, X.D. A Website Defacement Detection Method based on Machine Learning. In Proceedings of the International Conference on Engineering Research and Applications (ICERA 2018), Thai-Nguyen, Vietnam, 1–2 December 2018.
[2] X. Liang and J. Xu, "Control for networked control systems with remote and local controllers over unreliable communication channel," arXiv preprint arXiv:1803.01336, 2018.
[3] XueliHu, Qi Xi, and Zhenxing Wang. Monitoring of root privilege escalation in android kernel. In International Conference on Cloud Computing and Security, pages 491–503. Springer, 2018.
[4] EnamulKabir, JiankunHu, Hua Wang, GuangpingZhuo, A novel statistical technique for intrusion detection systems, Future Generation Computer Systems, 2017, ISSN 0167-739X.
[5] I. Ahmad, M. Basheri, M. J. Iqbal, and A. Rahim, ''Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection,'' IEEE Access, vol. 6, pp. 33789–33795, 2018.
[6] V. Patel, H. Madhukar, and S. Ravichandran, ''Variability index constant false alarm rate for marine target detection,'' in Proc. Conf. Signal Process. Commun. Eng. Syst. (SPACES), Jan. 2018, pp. 171–175.
[7] F. Farahnakian and J. Heikkonen, "A deep auto-encoder based approach for intrusion detection system," in Advanced Communication Technology (ICACT), 2018 20th International Conference on. IEEE, 2018, pp. 178–183.
[8] Wenjuan Li, Steven Tug, WeizhiMeng, and Yu Wang. Designing collaborative blockchained signature-based intrusion detection in iot environments. Future Generation Computer Systems, 96:481 – 489, 2019.
[9] R. Colelli, S. Panzieri, F. Pascucci, "Exploiting system model for securing cps: the anomaly based ids perspective", 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), vol. 1, pp. 1171-1174, Sep. 2018.
[10] S. J. Horng, M. Y. Su, Y. H. Chen, T. W. Kao, R. J. Chen, J. L. Lai and C. D. Perkasa, "A novel intrusion detection system based on hierarchical clustering and support vector machines," in Expert systems with Applications, vol. 38, no. 1, pp. 306–313, 2011.
[11] F. Amiri, M. R. Yousefi, C. Lucas, A. Shakery and N. Yazdani, "Mutual information-based feature selection for intrusion detection systems," in Journal of Network and Computer Applications, vol. 34, no. 4, pp.1184–1199, 2011.
[12] A. Gouveia and M. Correia, "Feature set tuning in statistical learning network intrusion detection," 2016 IEEE 15th International Symposium on Network Computing and Applications, 2016, pp. 68–75.
[13] S. O. Al-mamory and F. S. Jassim, "Evaluation of different data mining algorithms with KDD Cup 99 dataset," in Journal of Babylon University/Pure and Applied Sciences, vol. 21, no. 8, pp. 2663–2681, 2013.
[14] D. Deepika and V. Richhariya, "Intrusion detection withkNN classification and DS-Theory," in International Journal of Computer Science and Information Technology and Security, vol. 2, no.2, pp.274–281, 2012.
[15] B. Subba, S. Biswas and S. Karmakar, "Intrusion detection systems using linear discriminant analysis and logistic regression," 2015 Annual IEEE India Conference (INDICON), New Delhi, 2015, pp. 1-6.
[16] S. Devaraju and S. Ramakrishnan, "Performance comparison for intrusion detection system using neural network with KDD dataset," ICTACT Journal on Soft Computing, vol. 4, no. 3, 2014.
[17] A. A. Olusola, A. S. Oladele and D. O. Abosede, "Analysis of KDD'99 intrusion detection dataset for selection of relevance features," World Congress on Engineering and Computer Science, San Francisco, USA, 2010, vol. 1, pp. 20–22.
[18] R. Chitrakar and C. Huang, "Anomaly based intrusion detection using hybrid learning approach of combining k-medoids clustering and Na¨ıve Bayes classification," 2012 8th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM), Shanghai, 2012, pp. 1-5.