European Journal of Molecular & Clinical Medicine
ISSN 2515-8260 Volume 07, Issue 3, 2020

OPTIMIZED INTRUSION DETECTION CLASSIFICATION METHOD USING MACHINE LEARNING

RADHIKA D & M. DURAIRAJ

Radhika D, Research Scholar, Department of Computer Science and Engineering, Bharathidasan University, Tiruchirappalli – 620 023. Email: kvradhika2014@gmail.com.

Durairaj M, Assistant Professor, Department of Computer Science and Engineering, Bharathidasan University, Tiruchirappalli – 620 023. Email: durairaj.m@bdu.ac.in

ABSTRACT

Securing a network from the attackers is a challenging task at present as many users involve in variety of computer networks. To protect any individual host in a network or the entire network, some security system must be implemented. In this case, the Intrusion Detection System (IDS) is essential to protect the network from the intruders. The IDS have to deal with a lot of network packets with different characteristics. A signature-based IDS is a potential tool to understand former attacks and to define suitable method to conquest it in variety of applications. Data Mining techniques are used in the process of knowledge discovery for many domains' problems. Feature Selection plays a vital role for a large number of datasets. This paper discusses on the classification of attacks in the network with the assistance of the proposed Optimized Intrusion Detection Classification technique. In this proposed technique, the DBN hidden layers weights are optimized by using evolutionary Genetic algorithm. This GA is utilized to enhance the classification accuracy by applying the hidden layers of Restricted Boltzmann Machine (RBM). The comparative results show that the proposed classifier gives the improved accuracy, specificity, precision, Sensitivity, and reduced false positive rate, miss rate.

KEYWORDS: Machine Learning, Deep Belief Network, Genetic Algorithm, Intrusion Detection System, Classification

1. INTRODUCTION

The approaches for deflecting and recognizing the attacks are either a host-based or network-based IDS, which are the most customary IDS. The suspicious intent or malicious are specified by specific patterns and attack signatures are appeared in the product. When these patterns are appeared in the network traffic, then it is network-based IDS. If it is appeared in the log files, then it is log-based IDS. The IDS will be accurately active when it is comprised of both host-based IDS and Network-based IDS [1].

1.1 Network based Intrusion Detection System

The data source for Network-based IDS utilizes the raw packets. A network-based IDS typically utilizes a network adapter running in promiscuous mode to monitor and analyze all traffic in real-time as it travels across the network. This attack recognition module uses four common techniques to recognize an attack signature: (i) Pattern, expression or byte code matching, (ii) Frequency or threshold crossing, (iii) Correlation of lesser events, (iv) Statistical anomaly detection.

Once an attack has been detected, the IDS' response module provides a variety of options to notify, alert and take action in response to the attack. These responses vary by product, but usually involve administrator notification, connection termination and/or session recording for forensic analysis and evidence collection.

1.2 Host based Intrusion Detection System

Host-based intrusion detection started in the early 1980s before networks were as prevalent, complex and interconnected as they are today. In this simpler environment, it was common practice to review audit logs for suspicious activity. Intrusions were sufficiently rare that after the fact analysis proved adequate to prevent future attacks [2].

Today's host-based intrusion detection systems remain a powerful tool for understanding previous attacks and determining proper methods to defeat their future application. Host-based IDS still use audit logs, but they are much more automated, having evolved sophisticated and responsive detection techniques. Host based IDS typically monitor system, event, and security logs on Windows NT and syslog in UNIX environments. When any of these files change, the IDS compare the new log entry with attack signatures to see if there is a match. If so, the system responds with administrator alerts and other calls to action.

Host-based IDS have grown to include other technologies. One popular method for detecting intrusions checks key system files and executable via checksums at regular intervals for unexpected changes. The timeliness of the response is in direct relation to the frequency of the surveying interval. Finally, some products listen to port activity and alert administrators when specific ports are accessed. This type of detection brings an elementary level of network-based intrusion detection into the host-based environment.

2. RELATED WORKS

Chiba, Zouhair, et al[3]-[22] proposed to optimize a very popular soft computing tool widely used for intrusion detection namely, Back Propagation Neural Network (BPNN) using a novel hybrid framework (IGASAA)based on Improved Genetic Algorithm (IGA) and Simulated Annealing Algorithm(SAA). Genetic Algorithm (GA) is improved through optimization strategies, namely Parallel Processing and Fitness Value Hashing, which reduce execution time, convergence time and save processing power.

Lu, Huijuan, et al [4]-[37] proposed a hybrid feature selection algorithm that combines the mutual information maximization (MIM) and the adaptive genetic algorithm (AGA). Experimental results show that the proposing MIMAGA-Selection method significantly reduces the dimension of gene expression data and removes the redundancies for classification. The reduced gene expression dataset provides highest classification accuracy compared to conventional feature selection algorithms.

Vijayanand, R., D. Devaraj, and B. Kannapiran [5]-[41] proposed a hybrid genetic algorithm (GA) and mutual information (MI) based feature selection technique for IDS. The performance of IDS with the proposed feature selection techniques analyzed with IDS having mutual information, genetic algorithm and GA+MI based feature selection techniques using ADFA-LD dataset.

Shah, Asghar Ali, et al [6]-[47] proposed a model based on extensive survey to create an efficient hybrid classifier which is jointly based on feature selection, parameter optimization and classification. Feature selection is adapted to refine the area of interest by improving the accuracy of classification, then to optimize the parameters; genetic algorithm (GA) is the most appropriate technique to be used. Parameters optimization using GA also plays a remarkable role to improve classification using support vector machine (SVM).

Wang, Wei, Mengxue Zhao, and Jigang Wang [7]-[50] proposed a hybrid model based on deep auto encoder (DAE) and convolutional neural network (CNN). First, to improve the accuracy of malware detection, we reconstruct the high-dimensional features of Android applications (apps) and employ multiple CNN to detect Android malware. In the serial convolutional neural network architecture (CNN-S), a non-linear function, as the activation function to increase sparseness and "dropout" to prevent over-fitting. The convolutional layer and pooling layer are combined with the full-connection layer to enhance feature extraction capability.

3. GENETIC ALGORITHM

Genetic Algorithm is an intelligent probabilistic search algorithm which can be applied to a variety of combinational optimization problems. Theoretical foundations of Genetic Algorithm were initially developed by Holland in 1970's. The inspiration of GA is based on the evolutionary process of biological organisms in nature. During the course of evolution, natural population evolves according to the principle of natural selection and survival of the fittest. Individuals who are easily adaptable to all environmental conditions and have higher fitness are more likely to reproduce and generate offspring while lower fitness individuals are eliminated from population [8]-[16]. The combination of good characteristics from highly adaptive ancestors may produce even more fit offspring. In this way, species evolve more and more to become well adapted on environment.

A Genetic Algorithm stimulates these processes by taking an initial population of individuals and applying GA operators in each generation. Each individual is encoded as a chromosome which is a solution to the problem. A chromosome is a collection of genes, means an individual is made up of genes. The fitness of each individual is calculated by objective function. Highly fit individuals are given chances for reproduction, in crossover procedure. Mutation is optional for changing some of genes in individual to avoid duplicity. This evolution, selection, crossover process repeated until the condition is fulfilled.

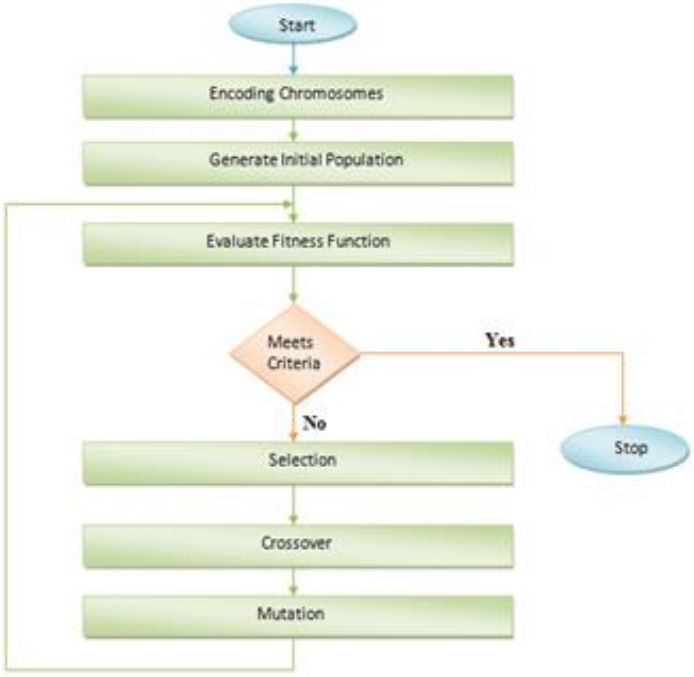

Figure 1: Flow of Genetic Algorithm

3.1 Encoding of a Chromosome

The chromosome should be encoded in such a way that it must represent information about the solution. The most commonly used way to encode chromosome is in binary string. Each bit represents some information about solution. Every chromosome is a collection of genes where each gene represents each bit of chromosome.

Table 1: Representation of the Chromosome

Chromosome 1	1110001101011010
Chromosome 2	1101010101110100

Pseudo Code of Genetic Algorithm

```
BEGIN
    Initialize population with random candidate Solutions
    Evaluate each candidate
    Repeat until (Termination condition is satisfied)
    DO
        Select parents
        Recombine pair of parents
        Mutate the resulting offspring
        Evaluate new candidate
        Select individuals for next generation
    END DO
END
```

Here the use of genetic algorithm is divided into two main sections: the pre-calculation section and also the detection section. In that pre-calculation section, collection of chromosome is created by set of training data. That chromosome sets are utilized in the next section for the purpose of comparison. The primary steps in pre-calculation as

Algorithm 1: Initialize chromosomes for evaluation

Input: Reduced Dataset (for training)

Output: A collection of chromosomes

Step 1: Range = 0.125

Step 2: For every training information

Step 3: If it has any nearest neighbour chromosome within Range

Step 4: Combine it with the adjacent chromosome

Step 5: Else

Step 6: Generate a new chromosome

Step 7: End if

Step 8: End for

4. DEEP BELIEF NETWORK

Deep belief networks (DBNs) [9][10]-[17][18] is a feasible multilayer neural system comprised of piling restricted Boltzmann machine for experiencing intricate data patterns. DBN can be distributed into two phases of the learning process. Initially, it is unsupervised learning that is employed by Contrastive Divergence (CD) algorithm called pre-training to acquire the initial weights from stacked RBMs. Secondly, it is supervised learning employed by error back-propagation(BP) algorithm named as fine-tuning that tunes the initial weights to get the final weights.

In Joint dissemination between visible layer x and hidden layer h gained by the energy-based probabilistic framework as follows:

$$p(x, h^1, \dots, h^n) = \left(\prod_{i=1}^{n-2} p\left(h^i \middle| h^{i+1}\right) \right) . p(h^{n-1} | h^n)$$

In the above equation, 1=h^0, $p\left(h^i \middle| h^{i+1}\right)$ is a conditional distribution in RBM for hidden-hidden units agreeing to the kth level of DBNs, and hidden joint distribution in top-level RBM is $p(h^{n-1}|h^n)$ in which each layer, the measured output of RBM is used as a subsequent layer input.

4.1 Restricted Boltzmann Machine

Restricted Boltzmann Machines (RBMs) [11]-[19] is the fundamental component of a DBNs that is preserved as a generative stochastic graphical and an unsupervised energy-based generative model. RBM is a level of visible units x and a layer of hidden units h, undirected and associated by proportionally weighted networks. Each hidden unit of RBM has the competence to encrypt at best one higher order interaction among inputs.

To epitomize the tricky complication, RBM entails a reduced amount of hidden units when an explicit amount of latent reasons given in the input and this state can be examined by RBM technique with contrastive divergence (CD) unsupervised learning algorithms. The technique elucidates the energy function as follow:

$$E(x, h; \theta) = -\sum_{i=1}^{l} \sum_{j=}^{m} x_i W_{ij} h_j - \sum_{i=1}^{l} b_i x_i - \sum_{j=1}^{l} a_j h_j$$

Where *l* and *m* are the number of visible and hidden units and θ = {w a b} are the model parameters. Specifically, the joint distribution of visible (x) and hidden (h) units are shown as follow:

$$P(x, h; \theta) = \frac{1}{Z(\theta)} \exp(-E(x, h; \theta))$$

In the above equation, Z(θ) is known as a partition function that is employed for stabilizing constant for energy function. And systematically computation of qualified prospects is as follow:

$$p\left(h_j = 1/x\right) = \sigma\left(b_j + \sum_{i=1}^{l} x_i w_{ij} \right)$$

$$p(x_i\, 1/h) = \sigma\left(a_i + \sum_{j=1}^{m} h_j w_{ji} \right)$$

Where σ is a sigmoid function.

5. PROPOSED OPTIMIZED INTRUSION DETECTION CLASSIFICATION (OIDC) METHOD

The following figure 2 gives the proposed classification method for categorizing the road accident dataset into Attack (YES), abnormal (NO) group. Deep Belief Network (DBN) has utilized for the arrangement and its hidden layer weights are augmented by using Genetic algorithm. The following phases are employed in this recommended classification method.

Step 1: Initialization of the Population

The personalities are produced unsystematically. For each genetic factor, the value of that gene is nominated as a random positive integer in the series, and the size of the initial population is 100.

Step 2: Function for Fitness and Objective

The root means a squared error of the weights is nominated as an objective function. The lesser value of the RMS of the weights means the better classification. The form of the objective function can be defined as follows:

$$g = \sqrt{\frac{1}{N}\sum_{t=1}^{N}|\ddot{x}_2|^2}$$

where N is the number of sampling points. In general, the elucidation epitomized by a chromosome is openly assessed by the fitness function. The fitness function is described by the (1/g). The individual with the greater fitness value displays that the property of control rules is healthier. When the advancement method finishes, the optimum individual can be acquired.

Step 3: Operation for Crossover

Crossover operation is the process of generating new individuals along with the boundary rate concluded the random combination of genetic materials opted from the parents. Two-point crossover algorithm is employed in this study. Each pair of parents has across over rate expressed by p_c= 0.95, and the offspring can be created by consuming it. Two points have unsystematically opted, and by associating the crossover rate and random rate, if the crossover condition is mollified, the new strings can be attained through consuming swap technique.

Step 4: Operation for Mutation

Here, in-order mutation technique is employed. Two mutation points have unsystematically opted and only the strings between these mutation points accomplish mutation. The mutation rate is p_m= 0.5. If the random rate is greater than the mutation rate, the strings between the points are in reciprocal order.

Step 5: Operation for Selection

Roulette wheel selection is a proportional selection operator. By using this technique, the fitness values are regularized via distributing the fitness value of each individual by the outline of the fitness values of all individuals. As a result, the probability distribution can be viewed as a roulette wheel, and the size of each slice is proportional to the normalized selection possibility of an individual. Selection can be equated to the circum-rotating of a roulette wheel and the slice which finishes up at the maximum will be documented. Subsequently, the resultant individual has opted. Roulette wheel selection methods employed and the selection rate is p_s= 0.95. The probability of the i^{th} individual is pronounced in the following expression:

$$P_i = \frac{f_i}{\sum_{i=1}^{I} f_i}$$

f_i is the fitness value of the ith individual. I is the size of all the individuals in the population space.

Step 6: Classification Stage

The working regulation of this phase hangs on the customary back-propagation algorithm. To identify and categorize the churners, non-churners and hesitant, an output layer is suggested as the maximum point of the Deep Neural Network. Furthermore, there is 'N' number of input neurons (based on the features), and three hidden layers are employed in the present investigation of Deep Neural Network. The optimized weight is calculated through the training stage with the support of a

training data set, where back propagation initiates with the weights that were accomplished in the pre-training stage. From the optimal weights, the layer work is revitalized and is presented as follows.

$$T(m_i = 1/n) = \sigma(m_i + \sum opt\_w_{ij} n_j)$$

$$T(n_i = 1/m) = \sigma(n_i + \sum opt\_w_{ij} m_j)$$

Where m and n represent the bias vector and hidden layers and σ is the sigmoid function with the range of (0,1). Moreover, the training dataset is accomplished in anticipation of the optimized weight is grasped, or determined accuracy is accomplished with the support of the above equation.

6. RESULT AND DISCUSSION

6.1 Description of the Dataset

Table 1 depicts the description of the dataset used in this paper. KDD CUP dataset [12] is utilized to find the intrusion in the network.

Table 1: Description of the KDD CUP Dataset

Sl.No	Feature Name	Feature Type
1	Duration	Continuous
2	Protocol_Type	Symbolic
3	Service	Symbolic
4	Flag	Symbolic
5	Src_bytes	Continuous
6	Dst_bytes	Continuous
7	Land	Symbolic
8	Wrong_fragment	Continuous
9	Urgent	Continuous
10	Hot	Continuous
11	Num_failed_logins	Continuous
12	Logged_in	Symbolic
13	Num_compromised	Continuous
14	Root_shell	Continuous
15	su_attempted	Continuous
16	num_root	Continuous
17	num_file_creations	Continuous
18	num_shells	Continuous
19	num_access_files	Continuous
20	num_outbound_cmds	Continuous
21	is_host_login	Symbolic
22	is_guest_login	Symbolic
23	count	Continuous
24	srv_count	Continuous
25	serror_rate	Continuous
26	srv_serror_rate	Continuous
27	rerror_rate	Continuous
28	srv_rerror_rate	Continuous
29	same_srv_rate	Continuous
30	diff_srv_rate	Continuous
31	srv_diff_host_rate	Continuous
32	dst_host_count	Continuous

33	dst_host_srv_count	Continuous
34	dst_host_same_srv_rate	Continuous
35	dst_host_diff_srv_rate	Continuous
36	dst_host_same_src_port_rate	Continuous
37	dst_host_srv_diff_host_rate	Continuous
38	dst_host_serror_rate	Continuous
39	dst_host_srv_serror_rate	Continuous
40	dst_host_rerror_rate	Continuous
41	dst_host_srv_rerror_rate	Continuous
42	Class	Normal & Attack

6.2 Performance Metrics

Table 2 depicts the performance metrics considered for evaluating the proposed classification method, Artificial Neural Network and Deep Belief Network.

Table 2: Performance Metrics

Metrics	**Equation**
Accuracy	$\frac{TP+TN}{TP+FN+TN+FP}$
True Positive Rate (TPR)	$\frac{TP}{TP+FN}$
False Positive Rate (FPR)	$\frac{FP}{FP+TN}$
Precision	$\frac{TP}{TP+FP}$
Specificity	1-FPR
Miss Rate	1- TPR

6.3 Performance Analysis of the Proposed Optimized Intrusion Detection Classification Method

Before the classification stage, feature engineering is done with the proposed Optimized Feature Selection method which combines the advantages of two optimization algorithms called FireFly algorithm and Animal Migration Algorithm. As a result of Feature Engineering, only 18 features are considered as the optimal feature subset. Table 3 depicts the features obtained in the feature engineering stage.

Table 3: Optimal Feature Subset obtained in the Feature Engineering stage

Sl.No	**Feature Name**
1	Duration
2	Protocol_Type
3	Service
4	Flag
5	Src_bytes
6	Dst_bytes
7	Land
8	su_attempted
9	num_shells
10	num_file_creations

11	num_root
12	count
13	num_access_files
14	srv_count
15	serror_rate
16	rerror_rate
17	dst_host_count
18	diff_srv_rate

Table 4 depicts the performance analysis of the proposed OIDC method, ANN and DBN classification technique using Original Dataset. The performance of the classifiers is evaluated with the table 2 performance metrics. From the table 4, it is clear that the proposed OIDC method gives accuracy of 83.45% than the other classifiers, TPR, Precision and Specificity is also increased by using proposed OIDC method than the DBN and ANN for original dataset. The error rates like FPR and Miss Rate is also minimized when using proposed OIDC method.

Table 4: Performance Analysis of the Proposed OIDC method, DBN and ANN using Original Dataset

Performance Metrics	**Classification Techniques**		
	Proposed OIDC Method	**DBN**	**ANN**
Accuracy	83.45%	76.40%	51.667%
True Positive Rate	0.782	0.675	0.517
Precision	0.718	0.6285	0.543
Specificity	0.548	0.438	0.259
False Positive Rate	0.452	0.562	0.741
Miss Rate	0.282	0.325	0.483

Table 5 depicts the performance analysis of the proposed OIDC method, ANN and DBN classification technique using Optimal Dataset. The performance of the classifiers is evaluated with the table 2 performance metrics. From the table 5, it is clear that the proposed OIDC method gives accuracy of 95.74% than the other classifiers, TPR, Precision and Specificity is also increased by using proposed OIDC method than the DBN and ANN for original dataset. The error rates like FPR and Miss Rate is also minimized when using proposed OIDC method.

Table 5: Performance Analysis of the Proposed OIDC method, DBN and ANN using Optimal Dataset

Performance Metrics	**Classification Techniques**		
	Proposed OIDC Method	**DBN**	**ANN**
Accuracy	95.74%	93.30%	92.3667%
True Positive Rate	0.9395	0.9042	0.892
Precision	0.962	0.898	0.896
Specificity	0.909	0.858	0.768
False Positive Rate	0.091	0.142	0.232
Miss Rate	0.0605	0.096	0.108

When comparing the table 4 and table 5, the proposed OIDC method performs better with optimal feature subset than the original dataset. Figure 2 depicts the graphical representation of the classification accuracy in (%) by the proposed OIDC method, DBN and ANN for the original dataset and optimal dataset. From the figure 2, it is clear that the proposed OIDC method gives more accuracy with optimal dataset than the other classification techniques.

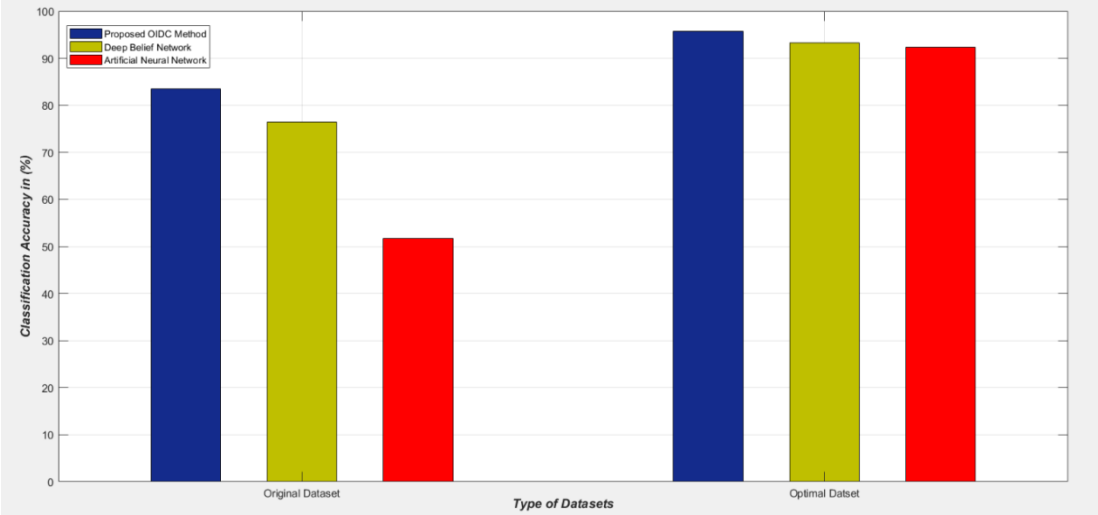

Figure 2: Graphical representation of the classification accuracy in (%) by the proposed OIDC method, DBN and ANN with original dataset and optimal dataset

Figure 3 depicts the graphical representation of the True Positive Rate (or Recall or Sensitivity) by the proposed OIDC method, DBN and ANN for the original dataset and optimal dataset. From the figure 3, it is clear that the proposed OIDC method gives more TPR with optimal dataset than the other classification techniques.

Figure 4 depicts the graphical representation of the precision by the proposed OIDC method, DBN and ANN for the original dataset and optimal dataset. From the figure 4, it is clear that the proposed OIDC method gives more precision with optimal dataset than the other classification techniques.

Figure 5 depicts the graphical representation of the specificity by the proposed OIDC method, DBN and ANN for the original dataset and optimal dataset. From the figure 5, it is clear that the proposed OIDC method gives more specificity with optimal dataset than the other classification techniques.

Figure 6 depicts the graphical representation of the False Positive Rate (FPR) by the proposed OIDC method, DBN and ANN for the original dataset and optimal dataset. From the figure 6, it is clear that the proposed OIDC method gives least FPR with optimal dataset than the other classification techniques.

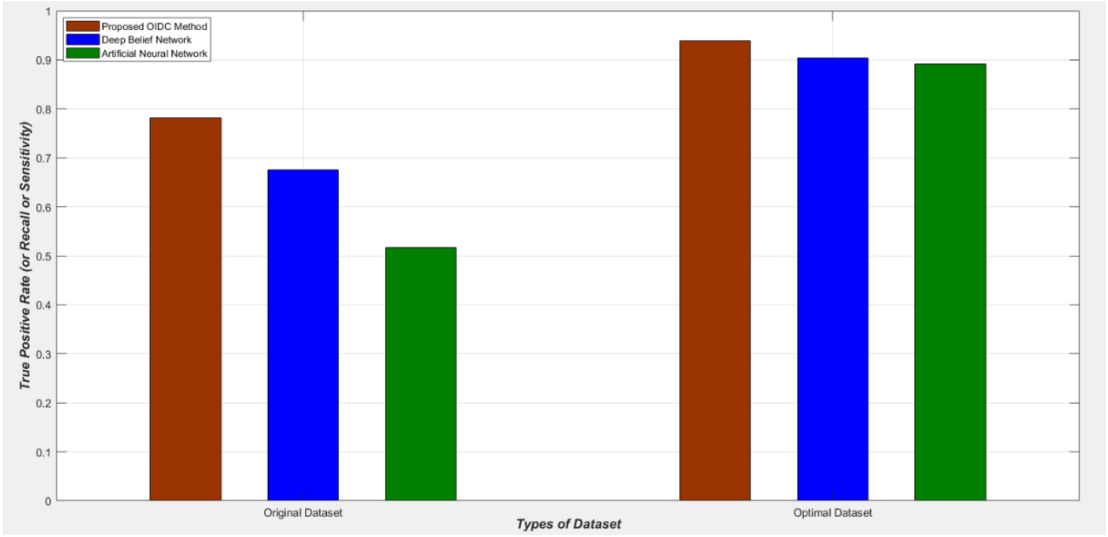

Figure 3: Graphical representation of the True Positive Rate (or Recall or Sensitivity) by the Proposed OIDC method, DBN and ANN with original and optimal datasets

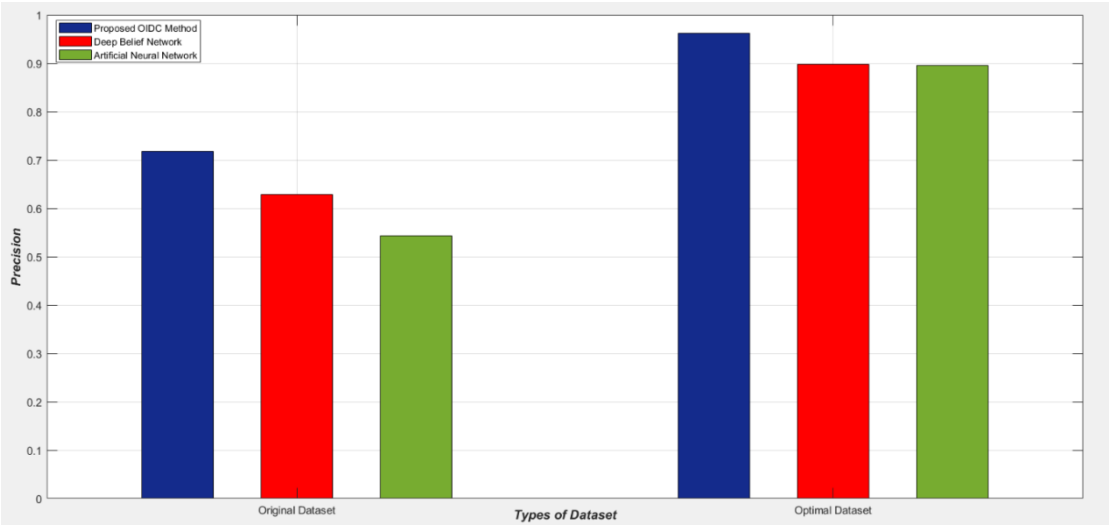

Figure 4: Graphical representation of the Precision by the proposed OIDC method, DBN and ANN with original and optimal dataset

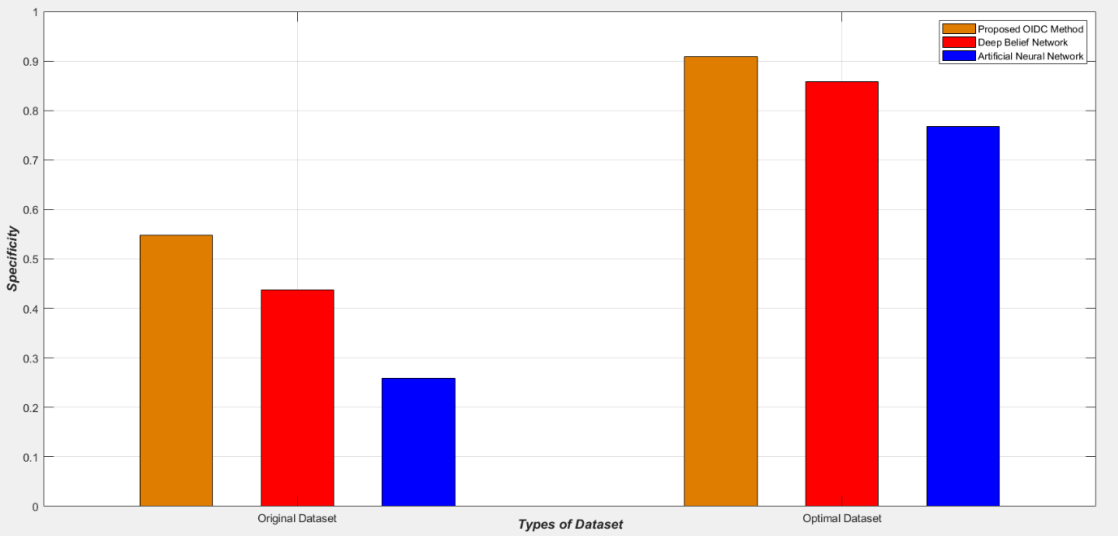

Figure 5: Graphical representation of the Specificity of the proposed OIDC method, Deep Belief Network and Artificial Neural Network with the original dataset and optimal dataset

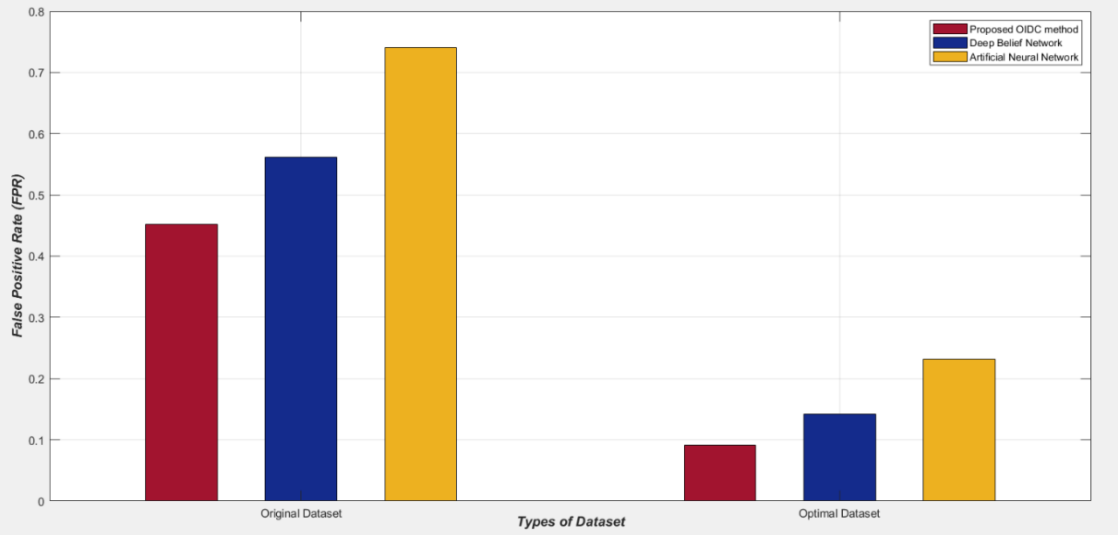

Figure 6: Graphical representation of the False Positive Rate (FPR) of the proposed OIDC method, Deep Belief Network and Artificial Neural Network with the original dataset and optimal dataset

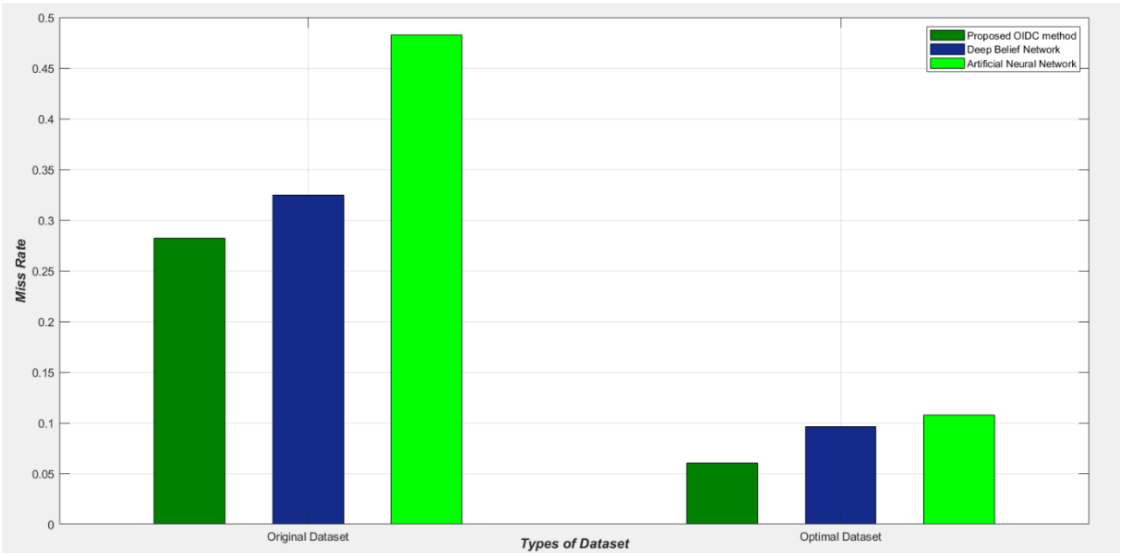

Figure 7: Graphical representation of the Miss Rate of the proposed OIDC method, Deep Belief Network and Artificial Neural Network with the original dataset and optimal dataset

Figure 7 depicts the graphical representation of the Miss Rate by the proposed OIDC method, DBN and ANN for the original dataset and optimal dataset. From the figure 7, it is clear that the proposed OIDC method gives least Miss Rate with optimal dataset than the other classification techniques.

7. CONCLUSION

In brief, the objective of this research study is to forecast the intrusion in the networks by using the proposed OIDC classification model. Through Deep Learning architecture and evolutionary algorithm, we planned to attain better accuracy in recognizing the attacks and abnormal nodes in the networks. Along with the experimental outcome, the proposed OIDC classification method is operative for the arrangement of attacks and abnormal in terms of accuracy, specificity, precision, sensitivity, False Negative Rate (Miss Rate) with its values. The precision level has visibly evident

that the recommended classification technique is deeply proficient in identifying the attacks in networks. The categorization presentations of this study establish the benefits of this technique: it is rapid, modest to operate, non-invasive

REFERENCES

[1] Han, Hong, et al. "Data mining aided signature discovery in network-based intrusion detection system." *ACM SIGOPS Operating Systems Review* 36.4 (2002): 7-13.

[2] Li, Wen, et al. "Context sensitive host-based IDS using hybrid automaton." *Journal of software* 20.1 (2009): 138-151.

[3] Chiba, Zouhair, et al. "A New Hybrid Framework Based on Improved Genetic Algorithm and Simulated Annealing Algorithm for Optimization of Network IDS Based on BP Neural Network." *The Proceedings of the Third International Conference on Smart City Applications*. Springer, Cham, 2018.

[4] Lu, Huijuan, et al. "A hybrid feature selection algorithm for gene expression data classification." *Neurocomputing* 256 (2017): 56-62.

[5] Vijayanand, R., D. Devaraj, and B. Kannapiran. "A novel intrusion detection system for wireless mesh network with hybrid feature selection technique based on GA and MI." *Journal of Intelligent & Fuzzy Systems* 34.3 (2018): 1243-1250.

[6] Wang, Wei, Mengxue Zhao, and Jigang Wang. "Effective android malware detection with a hybrid model based on deep autoencoder and convolutional neural network." *Journal of Ambient Intelligence and Humanized Computing* 10.8 (2019): 3035-3043.

[7] Demidova, L. A., M. M. Egin, and R. V. Tishkin. "A Self-tuning Multiobjective Genetic Algorithm with Application in the SVM Classification." *Procedia Computer Science* 150 (2019): 503-510.

[8] Demidova, L. A., M. M. Egin, and R. V. Tishkin. "A Self-tuning Multiobjective Genetic Algorithm with Application in the SVM Classification." *Procedia Computer Science* 150 (2019): 503-510.

[9] Verma, Maneesh Kumar, et al. "Phishing Website Detection Using Neural Network and Deep Belief Network." *Recent Findings in Intelligent Computing Techniques*. Springer, Singapore, 2019.293-300.

[10] Dan, Qiulin, et al. "A Fault Diagnosis Approach Based on Deep Belief Network and Its Application in Bearing Fault." *Proceedings of 2018 Chinese Intelligent Systems Conference*.Springer, Singapore, 2019.

[11] Jaworski, Maciej, et al. "Resource-Aware Data Stream Mining Using the Restricted Boltzmann Machine." *International Conference on Artificial Intelligence and Soft Computing*.Springer, Cham, 2019.

[12] https://archive.ics.uci.edu/ml/datasets/kdd+cup+1999+data