Innovative: International Multi-disciplinary Journal of Applied Technology
(ISSN 2995-486X) VOLUME 02 ISSUE 10, 2024

Anomaly Detection via Improved Rain Optimization Algorithm and Stacked Autoencoder Hoeffding Tree

Maryam Hsaini
Babylon Education Directorate/ Al-Baqer high school for girls

Firas Abdul Kadhim Mohamed Alhelli, Dalia Abdulrahim Mokheef
University of Babylon / College of Basic Education/Department of Mathematics

Abstract:

Professionals of Cybersecurity suggest that cyber-attack cost of damage would widely increase. Huge web usage increases stress over the way of securely passing on electronic info. Diagnosing faults and anomalies in real-time industrial systems is a concern because of enough covering an industrial system's complexity and difficulty. For security improvement, intrusion detection systems (IDSs) are applied for anomaly detection in network traffic. Now, IDS technology has concerns about performance based on false alarm notifications, unknown attack diagnoses, times, and accuracy detection. Machine learning (ML) methods have been increasingly applied in IDS for many years. However, such methods yet suffer from a lack of a labeled set of data, low accuracy, and heavy overhead. Reducing dimensionality plays an important role in IDS, as anomaly diagnosis from high dimensional network traffic attributes is a time-consuming process. Selection of Feature selection affects analysis speed. This paper concentrates on developing IDS effectiveness by applying the proposed Stacked Autoencoder Hoeffding Tree approach (SAE-HT) applying an Improved Rain Optimization Algorithm (IROA) for the selection of features. Experiments on the dataset NSL-KDD illustrate that our model multi-classification possesses great performance. In comparison to the other mechanisms of ML in the accuracy case, our model performs better than such mechanisms. The presented technique obtained an accuracy of 98.82 on a dataset of NSLKDD. Such outcomes of tests show that the presented strategy could properly and efficiently diagnose bad data.

Keywords: Anomaly Detection, feature selection, Rain Optimization Algorithm, Autoencoder, Hoeffding Tree.

1. Introduction

With internet technology improvement, a huge increase exists in info dimensions which are created, exchanged, and processed. Nearly in whole domains, a difficulty exists in controlling big high dimensional data amount. Such data become an aim for illegal tasks that impose a strict threat to the security of the network [1]. Despite different security apps' implementation like encryption, user authentication, malware prevention, and firewalls, a huge number of organizations suffer and fall victim to modern cyber-attacks. With the quick roll-out of info technology, attacks get complex quicker and accordingly potential threats would rise, so IDSs are defined for recognizing and reporting threats [2].

IDSs control a computer network for cyber-attacks. Traditional IDS methods depend on professionals of human subject matter to accurately generate signs that could properly diagnose cyber-attacks at the layer of the network. For over a decade studies have concentrated on developing IDS with machine learning (ML) techniques for reducing the overall need for human attempts. The majority of the present study has centered around misuse diagnosis and IDS based on ML is trained by applying the data set where whole cyber-attacks are labeled, its disadvantage which just the labeled attacks would be called to model, missing unfamiliar/novel attacks, basic data set labeling is a time-consuming and complicated activity prone to human error [3]. Based on the mechanism of detection, IDS could be grouped as misuse (known as sign-based) detection and anomaly (known as behavior-based) detection. Misuse detection strategies are modeled for attack detection attacks applying a predefined attack patterns database. They are increasingly efficient for known attack detection and preferred for the low false positive rate. So, they are not able to defend the system against unfamiliar attacks, due to that there are not any attacks in predefined pattern lists. Since network attacks go on to rise in frequency and diversity, keeping the updated database is time-consuming and impossible. In addition, misuse detection strategies could not detect zero-day attacks. In other words, unusual detection strategies utilize the usual system tasks for building normal-operation profiles, recognizing anomalies as manners that deviate from usual ones. These techniques are particularly appealing due to that they could potentially diagnose whole familiar and unfamiliar attacks kinds and zero-day attacks. However, basic anomaly detection strategies' demerit is that they need a tuning step and suffer from high false positive rates [4]. Datasets of Networks are sometimes high dimensional with several features. Many of the last research simply used whole features provisioned in data files. Although, based on whole features does not usually cause the best performance (for example, rate of detection); which refers to actual features subset might show the close/ better performance. Additionally, the invariant is that depending on the higher features number must impose the rising difficulty, so needing the wider time and sources amount to analyze [5]. Feature selection (FS) refers to the method, of relatively diagnosing features with higher importance and ignoring the less essential ones. The aim is to achieve a better set of features that could optimize, learn model complexity also define the provided issue accurately with min performance degradation.

Here, data mining is integrated with IDS for carrying out a particular activity. The activity is distinguishing essential, successfully covering up information in less time amount. This study concentrates on developing IDS effectiveness via applying the provided Stacked Autoencoder Hoeffding Tree approach (SAE-HT) utilizing an Improved Rain Optimization Algorithm (IROA) [6] for FS.

The present study begins with associated work for anomaly diagnosis applying methods of ML and their improvements in part 2. The method part defines model details created for anomaly detection in part 3. The present work follows details on the dataset applied for analysis in part 4. The paper provides the comparative multiple ML models vs deep learning models analysis in part 4. The result is given in part 5.

2. Related work

Broad research has been done on FS before. We summarize some that are nearly associated with the present work.

In [7], presents the hybrid machine-learning ensemble real-time anomaly-detection pipeline which integrates 3 ML models –Local Outlier Factor, One-Class SVM, and Autoencoder–, via the weighted medium for developing anomaly detection. The model of the ensemble used 3 air-blowing machines. This model novelty refers to that this includes 2 steps inspired by the standard industrial system: manufacturing and operation step.

In [8], defined CorrCorr, the technique of feature selection for multivariate correlation-based network anomaly detection systems. Assessed on UNSWNB15 and NSL-KDD IDS set of data, CorrCorr accordingly performed better than basic attributes and the ones which are chosen with a Principal Component Analysis (PCA) as well as the Pearson class label correlation. They analyzed the UNSW-NB15 dataset on feature correlations and recognized some faults.

In [9], apply Information Gain, ranking, and classifying attributes based on min weight amounts for choosing related and important attributes, after performing Random Tree (RT), J48 classifier algorithms, Random Forest (RF), Bayes Net (BN), Naive Bayes (NB) in tests on CICIDS-2017 dataset. Outcomes of tests illustrate that related and important features amount significantly shown by Information Gain influence detection accuracy and run time development.

In [10], the multi-step scheme for anomaly detection has been presented rectifying issues that happened in traditional DBSCAN. In the first provided solution step, the Boruta algorithm is applied to get related features' groups from a set of data. In the second step, the firefly algorithm, with the Davies-Bouldin Index-based K-medoid strategy, is applied for carrying out partitioning. In the third step, the locality-sensitive hashing based on the kernel is applied with traditional DBSCAN to solve the nearest neighbor search issue.

In [11], presents the efficient technique of DL, known as AE-IDS (Auto-Encoder Intrusion Detection System) given the algorithm of RF. The mentioned technique builds a set of training with feature selection and grouping. After training, such a model could predict outcomes with an auto-encoder that significantly decreases the time of detection and efficiently develops the accuracy of prediction.

In [12], applied ensemble technique as well as feature selection for developing ML-based IDSs performance. The basic reason was capitalizing on different algorithms' merits. In a dataset of CSE-CICIDS2018, ensemble learning was found to have higher detection accuracy than other single classifiers. Features of content were successfully eliminated from the set of data for developing performance. Spearman's rank correlation coefficient made a selection of 23 of 80 basic dataset features simple.

In [13], uses the Variational Auto-Encoder (VAE) feature representation capability in unsupervised anomaly detection and extracts worthy features for unmonitored anomaly detection functions. Tests of Comparison are performed on KDD CUP 99 and MNIST dataset. Outcomes illustrate that features achieved by VAE could make unmonitored anomaly detection strategies outperform. Auto-Encoder (AE) and Kernel Principle Component Analysis (KPCA) were used as comparisons. The outcome illustrates that VAE gets the best performance among them.

In [14], presented the novel algorithm of feature selection for feature selection applying the KDD CUP 99 dataset. Writers chose the best features from a total number of features (41) to detect intrusion in the network. Some techniques of feature selection, given the Mutual Information (MI) and wrapper with Bayesian network, C4.5 are applied for feature selection. With just the most suitable 10 features, the performance of detection is better than with 41 features and decreasing computational cost for the classifier. The efficiency of detection is developed with suitable features.

In [15], aims to answer the question of when autoencoders, a kind of semisupervised feedforward neural network, could present the low-cost anomaly detector technique for computer network data flow. Autoencoder techniques were assessed online with KDD'99 and UNSW-NB15 sets of data, showing that execution time and labeling cost are importantly decreased in comparison with traditional online classification methods for the same performance of detection.

In [16], provide the analysis UNSW-NB15 IDS set of data which would be applied to train and test our models. In addition, writers use a feature reduction method based on a filter applying the XGBoost algorithm. they perform the following ML strategies applying decreased feature space: KNN, SVM, Logistic Regression (LR), Decision Tree (DT), and Artificial Neural Network (ANN). In tests, they considered both binary and multiclass classification configurations. Outcomes illustrated which feature selection technique based on XGBoost lets techniques like DT raise the accuracy of the test for the binary classification model.

In [17], present a new 5-layer autoencoder (AE)-based scheme better suited for network anomaly detection functions. The proposal gives the outcomes that the writer achieved via the wide and rigorous investigation of some performance indicators included in the model of AE. In the provided model, they apply a novel data pre-processing method that converts and eliminates the most influenced outliers from input instances to decrease the scheme bias resulting from data imbalance across various kinds of data in a set of features. The presented model uses the most efficient reconstruction error task that plays an important role for a model for deciding if the network traffic instance is normal/anomalous.

In [18], presents an outlier-detection algorithm to detect network traffic anomalies given the clustering algorithm and the model of the autoencoder. The algorithm of BIRCH clustering is used as an autoencoder pre-algorithm for pre-classifying sets of data with complicated data distribution features when an autoencoder scheme is applied for outliers detection given the threshold. Such outcomes of the test show that the presented strategy could efficiently and properly diagnose anomalous data.

3. Proposed method

The presented work utilizes a benchmark dataset of NSL_KDD for analysis of intrusion. the first step concentrates on the selection of features by applying the Improved Rain Optimization Algorithm and chooses the main features that contribute to intrusion. The second step focuses on using the presented method of Stacked Encoding Hoeffding Tree for classifying data given the metrics of performance such as sensitivity, accuracy, false-negative rate, F1 score, specificity, and false-positive rate.

The Benchmark dataset of NSL_KDD is considered for analysis. The Set of data that we have considered is a standalone dataset for incorporating data of stream, a set of data is streamed by applying methods in the Matlab tool. The method of system object simplifies the process of streaming in Matlab. Now, data is continuous and that has streaming data features. attacks are detected applying the presented classification strategy of Stacked Autoencoder Hoeffding Tree (SAE-HT). A bio-inspired method known as the Improved Rain Optimization Algorithm (IROA) increases SAE-HT classification method performance. distracting variance is eliminated from data by applying the method of IROA feature selection which makes the classifier able to outperform, particularly while coping with high dimensional features. Fig. 1 illustrates the SAE-HT classification technique flow method.

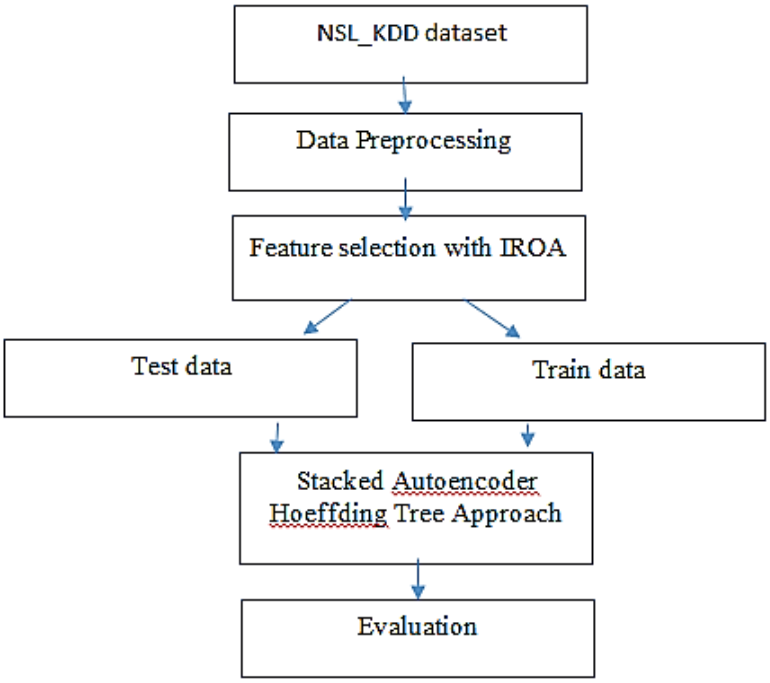

Figure 1. Flowchart of SAE-HT classification technique.

3.1. Data Preprocessing

Data Preprocessing is an important step in ML. Raw data collected is made ready to be applied by ML techniques to extract meaningful insights from data. NSL_KDD Dataset taken from the Canadian Institute for Cybersecurity is analyzed. While the downloaded dataset is in gz /Tcpdump form, shift that to CSV file format and dataset load in the environment. The presented method of classification just supports numeric data. As most of the methods of machine learning (ML) apply mathematical equations that just support numeric data use, categorical data conversion in numerical data applying functions of data conversion must happen. one-hot method of encoding changes categorical data to numerical data, hence making the easy-to-use methods of machine learning to set of data.

3.1.1. One-hot encoding technique

One-hot encoding technique is the most effective encoding technology to deal with several conversions to categorical attributes. This could shift categorical features to binary vectors. vector holds Zeroes, One's as amounts. vector holds only 1 element with amount 1 and the other amounts associated with Zero. Element with amount 1 illustrates possible amounts happening against categorical attributes. The dataset of NSL_KDD includes 3 categorical features flag, protocol_type, and service. For instance, protocol_type includes 3 features: UDP, ICMP, and TCP. By using one hot encoding technique, ICMP could be encoded as (1,0,0), TCP could be encoded as (0,1,0), and UDP could be encoded as (0,0,1). Similarly, categorical features' service and flag are encoded in one-hot encoding vectors.

3.2. Feature Selection with Improved Rain Optimization Algorithm

Selection of Feature decreases features by eliminating less significant/ insignificant cases. A lot of feature selection methods exist. The study applies the method of Bio-Inspired feature selection known as an Improved Rain Optimization Algorithm (IROA) for choosing optimum features. The basic IROA aim is finding non-redundant and highly correlated features, therefore removing the least correlated features.

3.2.1. Improved Rain Optimization Algorithm (IROA)

While rainfall begins; droplets fall onto the surface of the earth. After a while, droplets are linked to each other and this combines and moves on a weight-based surface. The other droplets go to consecutive droplets and are linked with them and some may be tired/monitored by soil on soil features base such as the texture of permeability, soil surface, porosity, wettability, etc. Furthermore, few soils are not solved in water that droplets dropped on the fat area are fully attracted by the soil and this has vanished from the inclined area and joins the other droplets to improve the flow. Some flows are joined with each other and convert into rivers. If any barrier inflows/ rivers way, lakes are improved that water quantity shows droplets importance. When rain is stopped, streams and rivers are discharged to local lakes. Then, small lakes don't appear due to water evaporating in a lake/ this is attracted by soil. So, essential lakes are sustained in sets based on the earth's surface and soil feature topology. These lakes show local minima of the earth's surface and deeper lakes show global minima. While the kind of rain is converted, a small change exists in a predefined way. For instance, when heavy rain with great droplets, each droplet is robustly joined to each other without any absorption/exhaustion in food. Now, global in is predicted and local min is joined to each other due to rainstorm. Accordingly, when a light rain exists with smaller droplets, each droplet is attracted by soil causing no improvement of flow. So, this is analyzed which parameter tuning has crucial importance in using ROA time [19].

Particle movement in the provided scheme is similar to gradient-related optimization strategies and traditional single-point schemes such as hill-climbing (HC) and gradient-descent (GD), rainfall optimization algorithm (RFO). Such strategies modify the unique parameter for whole iterations to recognize if the setting of this parameter increases task cost. Thus, ROA uses the answers' set that moves to the best one simultaneously that such features change in whole iterations.

Algorithmic steps of ROA

Now, the manner of rain is simulated as this is described in the traditional part. Each solution to the issue can be like a raindrop. According to such problems, some points in the space of the answer are decided in a random model as raindrops fall on the surface of the earth. The main raindrop feature refers to the radius. The whole raindrops radius can be restricted as time passes and this is increased as the raindrop is joined to alternate drops. When a basic answer population is created, each droplet radius is assigned in arbitrary behavior to a restricted extent. In addition, every droplet confirms the neighborhood given the size. Single droplets that are not joined still confirm the end place limit that has been covered. For resolving problems in dimensional space, every droplet is included in the n variable. So, in the basic stage, the max and min variable limit is confirmed since restrictions are calculated by droplet radius. Then, two variable endpoints are sampled and this is repeated until obtaining the last criteria. After that, the first cost droplet is upgraded by moving in a downward direction. This is carried out for every droplet, cost, and each droplet position will be assigned. The droplet radius will be changed in two models.

While two droplets with radius r1 and r2, are nearer to each other with the normal region and this joins to improve bigger radius R droplet [19]:

$$R = \left(r_1^n + r_2^n\right)^{1/n}, \tag{1}$$

That n shows the variables' count for whole droplets. If the droplet with radius r1 is not moved, given the soil features which are illustrated by α, water is monitored by soil [19].

$$R = (\alpha r_1^n)^{1/n}. \tag{2}$$

Clearly, α shows the droplet amount that has been attracted in whole iterations from 0 to 100%. In addition, this defines the least amounts for droplets radius rmin, that droplets with min radius of that rmin will be diminished.

Improved IROA rain algorithm

In the ROA rain optimization algorithm, first, the location of the raindrops is chosen randomly, and then each raindrop moves towards the minimum points during a process, finally, as the raindrop becomes smaller, the accuracy of the answer increases in each iteration. The negative point is that There was in this algorithm, sometimes causes the lack of a finder with a reasonable cost, is that the diameter of the raindrops may become very small very quickly and before we reach the desired extreme region, the number of NFE[1] increases greatly. Also, in many cases, the raindrop has to travel a very long way in the form of a turtle to the desired area, so to solve this problem, we decided that the randomly selected raindrop should first perform a general search and then the ROA method. Let's look for the exact value of the answer. According to others, the algorithm will search in two ways for each raindrop in each repetition [6]:

1. General search

2. Local search

The local search is the same as the rain algorithm, but the general search is performed as follows. It becomes a certain number of smaller drops and disperses in the search space; Just like a cluster bomb going off at one point. In this case, several smaller bombs are randomly scattered around the main bomb. How each raindrop is divided into several smaller drops and how far from the main drop it is scattered will be among the input parameters of the network. In this way, in the first step, each raindrop is converted into several smaller drops that are scattered around it, and the value of the objective function is calculated for each of the smaller drops, and the drop that has the smallest value of the objective function is replaced with the original drop. This drop moves under the rain algorithm and reaches a better point. The same process happens for the rest of the raindrops until we reach the last drop in the iteration.

Next, the same scenario will be done again. That is, first a general search and then a local search will be performed. Therefore, in each iteration, we face two radii, the general search radius (R) and the raindrop radius (R), both of which can decrease or remain constant in each iteration. It is recommended to choose the value of R_1 equal to half of the search range in the first iteration [6].

$$R_1 = \frac{\text{var}_{\max} - \text{var}_{\min}}{2} \tag{3}$$

If the algorithm succeeds in improving the location of the raindrop while performing a general search for a raindrop in one iteration, the value of R1 can be reduced so that the radius of the general search becomes narrower in the next iteration, and this will improve the final results [6].

$$R_1 = \alpha R_1 \tag{4}$$

In this research, the value of α is considered equal to 0.8.

4. Result Analysis

4.1. Data set description

The presented study applies a benchmark dataset of NSL_KDD for analysis of intrusion. The dataset of KDD CUP'99 is a well-known dataset of benchmarks applied for a system of network intrusion detection. The basic KDD CUP'99 dataset restriction is that includes high extra record numbers which have an impact on evaluated system effectiveness. The developed KDD CUP'99

[1] Number of Function Evaluation

version refers to a dataset of NSL_KDD where extra records are eliminated. The dataset of NSL_KDD has nearly 125,973 data of training and 22,544 data of testing. Like KDD CUP'99, records in a dataset of NSL KDD are singly labeled as anomaly and normal. That has 41 attributes that address 4 various attack groups [20].

4.2 Experimental Setup

The ability of a test to correctly distinguish fluctuations and normal events from others is called accuracy. For computing test accuracy, one should get true positive and negative instances' total ratio to the sum cases' number. Mathematically, the rate could be assessed as:

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (5)$$

Sensitivity is called as rate of true positive. Sensitivity is described as:

$$sensitivity = \frac{TP}{TP+FN} \quad (6)$$

$Specificity$ is described as:

$$specificity = \frac{TN}{TN+FP} \quad (7)$$

The variable of F-score integrates 2 two precision and sensitivity variables and is described as:

$$\text{F-score}=\frac{2\times precision\times sensitivity}{precision+sensitivity} \quad (8)$$

In addition to the mentioned criteria, the terms N, P, TP, TN, FP, and FN are summarized in a matrix called the confusion matrix, which is shown in the figure below.

Table 1- The confusion matrix

Predicted class				
Real class		yes	no	total
	yes	TP	FN	P
	no	FP	TN	N
	total	P'	N'	P+N

However, the mentioned confusion matrix is designed for datasets with two classes, it can easily and similarly be generalized for data with more than two class labels.

We compared our presented model performance with other similar models (articles [17] and [18]). We compared performance by applying 4 metrics, known as F1-score, accuracy, sensitivity, and specificity. Table 2 shows that our presented technique could get an accuracy higher than 98.8% and the highest F1-score of 99.31%. From Table 1, we can observe that our outcomes show that our strategy suggests high classes of f1-score, precision, accuracy, and recall particularly when we apply IROA for the selection of Features.

Table 2. Performance comparison with other approaches on NSLKDD

Method	Accuracy	Precision	Sensitivity	Specificity	F1-score
Paper [17]	90.61	86.83	98.43	-	92.26
Paper [18]	87.88	89.81	88.05	-	88.46
Proposed method	98.82	99.33	99.29	99.28	99.31

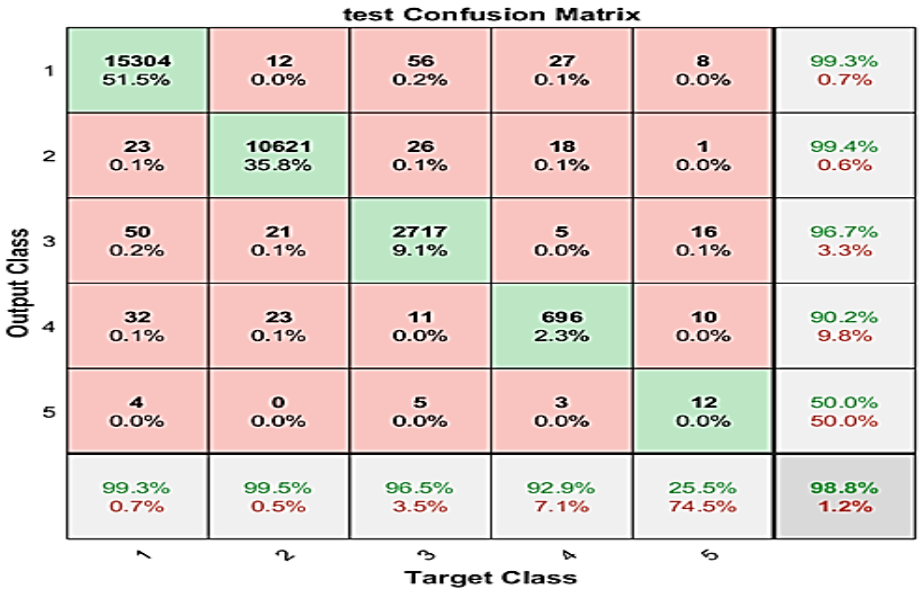

Fig. 2: Confusion Matrix of NSL-KDD Data set using the proposed method.

Fig. 2 illustrates the NSLKDD dataset confusion matrix by applying DNN. It shows that the whole accuracy of IDS is 98.8%. The outcomes of the test illustrate that most instances are focused on the confusion matrix diagonal showing that the whole performance of classification is very high. Although this could be intuitively observed from the confusion matrix in Fig. 2 illustrates that the presented technique obtains great detection performance for allocating usual traffic from attack ones, however further development is yet to exist for allocating various attack traffic. Dos and Probe attack traffic detection influence are relatively great when R2L and U2R attack traffic is not valid.

ROC curve is given the confusion matrix that is based on assessing model prediction ability. In certain sets of data, a group imbalance event exists. More negative instances exist than positive ones (vs.). positive and negative instances shared for checking the dataset might convert with time. So, the ROC curve could stay without change.

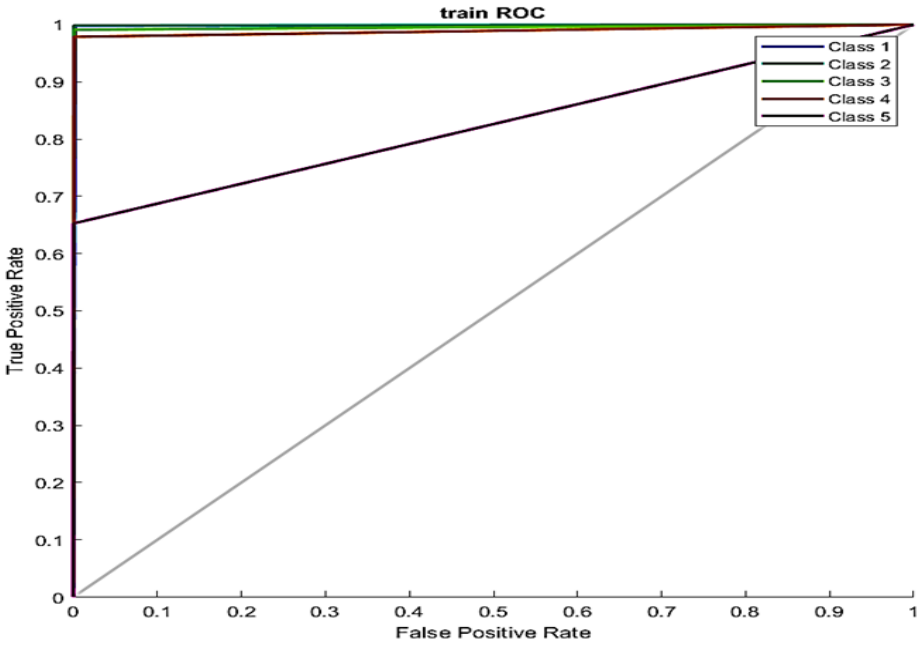

Fig. 3: ROC Curve of NSL-KDD Data set using the proposed method

Figure 3 shows our proposed model curve of ROC (receiver operating feature curve). Our model adopts great performance by generating a score of AUC (area under the curve of ROC) with a high rate of true positive rate with a rate of low false-positive.

5. Conclusion

Unusual detection of a Network plays a vital role as that presents an efficient algorithm for preventing cyberattacks. With new Artificial Intelligence (AI) improvement, several deep learning strategies based on Autoencoder (AE) exist for network anomaly detection for developing our posture to the safety of the network. Present new AE models' performance applied for anomaly detection of network differs with no holistic strategy proposed for comprehending crucial AE models and detection accuracy essential performance indicators core set effects. This paper concentrates on developing IDS effectiveness by applying the proposed Stacked Autoencoder Hoeffding Tree approach (SAE-HT) applying an Improved Rain Optimization Algorithm (IROA) for the selection of feature. Experiments on the dataset NSL-KDD illustrate that our model multi-classification possesses great performance. In comparison to the other mechanisms of ML in accuracy cases, our model performs better than such algorithms. Our future study would be directed to checking deep learning as a device of feature extraction for learning effective presentation of data for the issue of anomaly detection.

References

1. Seraphim, B. Ida, E. Poovammal, Kadiyala Ramana, Natalia Kryvinska, and N. Penchalaiah. "A hybrid network intrusion detection using darwinian particle swarm optimization and stacked autoencoder hoeffding tree." Math. Biosci. Eng 18 (2021): 8024-8044.
2. Doreswamy, Mohammad Kazim Hooshmand, and Ibrahim Gad. "Feature selection approach using ensemble learning for network anomaly detection." CAAI Transactions on Intelligence Technology 5, no. 4 (2020): 283-293.
3. Nixon, Christopher, Mohamed Sedky, and Mohamed Hassan. "Salad: An exploration of split active learning based unsupervised network data stream anomaly detection using autoencoders." (2021).
4. Pu, Guo, Lijuan Wang, Jun Shen, and Fang Dong. "A hybrid unsupervised clustering-based anomaly detection method." Tsinghua Science and Technology 26, no. 2 (2020): 146-153.
5. Nakashima, Makiya, Alex Sim, Youngsoo Kim, Jonghyun Kim, and Jinoh Kim. "Automated feature selection for anomaly detection in network traffic data." ACM Transactions on Management Information Systems (TMIS) 12, no. 3 (2021): 1-28.
6. Nouri, Hojjat, and Ali Anuri. "Using an improved rain optimization algorithm to simulate the movement of fluid slurry in the fracture and matrix." Oil Research 33, no. 1402-2 (2023): 20-39
7. Velásquez, David, Enrique Pérez, Xabier Oregui, Arkaitz Artetxe, Jorge Manteca, Jordi Escayola Mansilla, Mauricio Toro, Mikel Maiza, and Basilio Sierra. "A hybrid machine-learning ensemble for anomaly detection in real-time industry 4.0 systems." IEEE Access 10 (2022): 72024-72036.
8. Gottwalt, Florian, Elizabeth Chang, and Tharam Dillon. "CorrCorr: A feature selection method for multivariate correlation network anomaly detection techniques." Computers & Security 83 (2019): 234-245.
9. Stiawan, Deris, Mohd Yazid Bin Idris, Alwi M. Bamhdi, and Rahmat Budiarto. "CICIDS-2017 dataset feature analysis with information gain for anomaly detection." IEEE Access 8 (2020): 132911-132921.

10. Garg, Sahil, Kuljeet Kaur, Shalini Batra, Georges Kaddoum, Neeraj Kumar, and Azzedine Boukerche. "A multi-stage anomaly detection scheme for augmenting the security in IoT-enabled applications." Future Generation Computer Systems 104 (2020): 105-118.
11. Li, XuKui, Wei Chen, Qianru Zhang, and Lifa Wu. "Building auto-encoder intrusion detection system based on random forest feature selection." Computers & Security 95 (2020): 101851.
12. Fitni, Qusyairi Ridho Saeful, and Kalamullah Ramli. "Implementation of ensemble learning and feature selection for performance improvements in anomaly-based intrusion detection systems." In 2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), pp. 118-124. IEEE, 2020.
13. Yao, Rong, Chongdang Liu, Linxuan Zhang, and Peng Peng. "Unsupervised anomaly detection using variational auto-encoder based feature extraction." In 2019 IEEE International Conference on Prognostics and Health Management (ICPHM), pp. 1-7. IEEE, 2019.
14. Selvakumar, B., and Karuppiah Muneeswaran. "Firefly algorithm based feature selection for network intrusion detection." Computers & Security 81 (2019): 148-155.
15. Nixon, Christopher, Mohamed Sedky, and Mohamed Hassan. "Autoencoders: a low cost anomaly detection method for computer network data streams." In Proceedings of the 2020 4th International Conference on Cloud and Big Data Computing, pp. 58-62. 2020.
16. Kasongo, Sydney M., and Yanxia Sun. "Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset." Journal of Big Data 7 (2020): 1-20.
17. Xu, Wen, Julian Jang-Jaccard, Amardeep Singh, Yuanyuan Wei, and Fariza Sabrina. "Improving performance of autoencoder-based network anomaly detection on nsl-kdd dataset." IEEE Access 9 (2021): 140136-140146.
18. Wang, Dongqi, Mingshuo Nie, and Dongming Chen. "BAE: Anomaly Detection Algorithm Based on Clustering and Autoencoder." *Mathematics* 11, no. 15 (2023): 3398.
19. Pustokhina, Irina V., Denis A. Pustokhin, Phong Thanh Nguyen, Mohamed Elhoseny, and K. Shankar. "Multi-objective rain optimization algorithm with WELM model for customer churn prediction in telecommunication sector." Complex & Intelligent Systems (2021): 1-13.
20. Tavallaee, Mahbod, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani. "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. Ieee, 2009.