PREDICTING THE TRACES OF ADVERSARIES USING THE ENHANCED DATA ENCODING METHOD

R. Santhosh

Abstract

Network intrusion detection research has become a major research area for deep learning (DL). To significantly increase detection performance and accuracy, we provide an upgraded learning network-based intrusion detection model known as Enhanced Data Encoder (EDE). This approach can result in challenges, including the model converging to local optima or requiring extended training durations. To overcome these challenges, we propose an EDE with the supervised learning-capable model. To solve the problems of poor classification performance, which often arise from EDE random kernel parameter initialization is proposed. Experimental evaluations conducted on the BoT-IoT, IoT Network, MQTT, IoT-23, IoT-DS-1, and IoT-DS-2 datasets demonstrate that the proposed EDE algorithm performs better than existing approaches in terms of accuracy and other evaluation metrics.

Keywords :

Intrusion detection, deep learning, prediction, training, and testing

I. INTRODUCTION

Network technologies like 5G, cloud computing and IoT have developed so quickly where networks are producing huge data. This surge poses significant challenges and obstacles for network security, an area of research that is gaining prominence [1]. A key technique within this domain is intrusion detection (ID), which encompasses two primary tasks: 1) Examining current network data to determine and describe attack data characteristics, then comparing them to host or network data. We call this procedure "misuse detection." 2) The establishment of a connection between the widespread patterns of behavior found in a sample database and the network data [2]. Intrusion behaviors are defined as any departures from the behavior traits. This procedure is called anomaly detection. Despite having a low false alarm rate and being successful in recognizing the behavioral characteristics of known data breaches, misuse detection is unable to adjust or pick up on new threats [3]. As a result, in order to adjust to the ever-changing environment, the matching database needs to be updated often. Furthermore, abuse detection typically performs poorly when it comes to identifying new attacks. However, anomaly detection has a significant false alarm rate despite its ability to detect unknown attacks [4].

Unusual network traffic is stopped after it has been discovered in the ID system. Following the identification of the network attack type, the attack type's characteristic database is continuously enhanced, strengthening the system's defenses. In this sense, binary class classification and multi-classification issues are related to network anomaly detection. Researchers have employed data mining, ML, and NNs to discover anomalies in networks in recent years, with encouraging outcomes. However, feature extraction (FE) [5] and data selection are key components of data mining and classical ML techniques, which frequently fail to produce adequate results. As a result, considerable improvement is required in the data processing methods for, for example, the precision and accuracy of categorization. According to [6], a network model utilizing a deep neural network (DNN) architecture is capable of identifying more significant data elements within a dataset. In 2006, Hinton successfully employed a DL model to effectively reduce and classify the MNIST dataset. Furthermore, DL has showcased substantial potential in the area of data classification, achieving notable advancements across various fields, including emotional analysis, control systems and natural language processing [7].

In conclusion, the aforementioned research findings have three drawbacks. Firstly, real-time network traffic detection and feedback, latency reduction, and detection efficiency are all requirements for ID systems. Second, there is a lot of space for improvement in ID's generalization and accuracy [8]. An effective model can enhance the ID system's performance and identify additional kinds of attacks. The creation of ID models will be impacted by attack data that is incorrectly classified. Thirdly, most of the research listed above apply when the kind of network traffic is understood. Traditional classifiers frequently make poor decisions in the face of unknown threats, which impairs or even deteriorates

Department of Computer Science Engineering
Karpagam Academy of Higher Education
Coimbatore, India
santhoshrd@gmail.com*

* Corresponding Author

the effectiveness of ID systems [9]. The ID system's capacity for learning becomes especially crucial at this point. It must be able to identify unidentified attacks [10]. To address the problems listed above, this study suggests the enhanced data encoder (EDE) prediction model, which makes use of the benefits of DNNs in the identification field. To enhance DBN's performance, the EDE algorithm is utilized where the traditional approach for classification after the dimensionality reduction of high-dimensional data features by the EDE model. To improve the classifier's generalization across different datasets, we use the EDE to optimize the parameters. The model optimizes training and detection time while also considerably improving classification precision, TPR, and accuracy, according to experimental results across several datasets.

The remainder of this paper is organized as follows: section 2 explores the review and advancements of EDE architecture is given in section 3. The metrics are practically analyzed in section 4. The work summary is given in section 5.

II. RELATED WORKS

One common benchmark dataset used to boost intrusion detection (ID) system performance is KDDCup99. Developed in 1998 by the DARPA Intrusion Detection Evaluation Network, during the Data Mining Tools and Third International Knowledge Discovery Competition [11], this dataset was utilized. Creating a prediction model that could distinguish between two types of network connections attack connections and regular connections was the main goal of this project. Attacks can take several forms, such as U2R, R2L, DoS and Probe. The 41 characteristics were created using the automated model for IDS framework for feature development in the KDDCup99 competition [12]. The package's basic characteristics are the first nine; content characteristics are the 10-22 group; traffic characteristics are the 23-31 group; and host-based characteristics are the 32-41 group. There are two types of datasets available: The entire dataset, along with an additional 10% of complementary data has been utilized [13]. Since that time, the competition has continued to serve as a baseline job. Even though many ML approaches have emerged since this competition began, only a small number of specific datasets have been used, and most published results only use 10% of the training and testing data. A comprehensive literature review focusing on ID-based ML has recently been conducted using the KDDCup99 dataset.

According to a study referenced in [14], the most effective detection rate approach is the ID model utilizing shared nearest neighbors (SNN). The fundamental characteristics of the package are identified as the first nine, while content characteristics are classified as numbers 10 through 22. By compressing the dataset, SNN was determined to perform well in the K-means clustering of the U2R attack category; however, no results were obtained when applied to the full test dataset. Utilizing a naive Bayesian network, [15] indicates related attributes with leaf nodes and connected classes with root nodes. Additionally, by modeling temporal and spatial information, an evolutionary algorithm based on Network Intrusion Detection Systems (NIDS) is suggested that can detect complex irregular patterns. The application of swarm intelligence technologies, including particle swarm optimization, ant colony clustering, and ant colony optimization is explored in [15], while integrated learning technologies are assessed. The package's basic characteristics are the first nine; content characteristics are the 10-22. In these kinds of studies, a comparative analysis reveals that descriptive statistics were primarily employed. A thorough analysis of the literature reveals that, generally speaking, not many research employ modern DL techniques for NIDS. KDDCup99 and NSL-KDD are the benchmark datasets for experimental analysis that are frequently used [15].

III. METHODOLOGY

A. Pre -processing

The accuracy rate will be lower if missing data is used directly for categorization. Based on the information provided, the k-means clustering technique is used in this study to more accurately handle the problem of missing data. To eliminate the sample, the system has been improved. It's important to note that the differing sizes and units of measurement of various evaluation criteria can sometimes impact the outcomes of medical data studies. Our study use this technique to counteract this by eliminating dimensional interactions across indicators and considering how additional medical data affects the definition of parameters. To standardize the data, we also employ the Z-score normalization technique. Data restoration is the process of retrieving backup data from secondary storage and transferring it to a new location or back to its original location. This procedure allows data recovery and restoration to its original state in situations where information has been lost, stolen, or corrupted. This process is used to normalize the data and ensure that the updated data has a normal distribution. It also provides the mean and SD of the original dataset. The results demonstrate that the transformation functions for the two variables are identical, having the mean of 0 and SD of 1.

$$X^* = \frac{x - \mu}{\sigma} \qquad (1)$$

A number of important factors are taken into account in this study, such as σ, the sample standard deviation, and μ, the overall average of the sample data. It is clear that there is a significant skew in the distribution of disease data, with a noticeably smaller number of categories representing cases of illness than those representing normal circumstances. During the pre-processing stage, this work used SMOTE to address the data imbalance problem. It is crucial to highlight the distinctions between the novel strategy we suggest and the conventional over-sampling technique, which merely duplicates minority class data. Our method applies linear interpolation (LI) techniques to the existing minority class samples to improve minority class instance production. AE is a strong version of the auto-encoder. The fundamental idea is to carefully introduce random noise into the initial input data, causing it to become corrupted. The auto-encoder must then be trained to retrieve the original data from this modified input. The AE structure is depicted in Fig 1. Let $P = f(\theta M + \beta)$ after the input M is distorted by noise to form $\overline{M}$. The hidden layer specifies to P for encoding. This layer seeks to extract the original input's attributes, such as the deviation vector β and the learning mapping matrix θ. The decoder component reconstructs the original data and reduces construction error. Configuration parameters to maximize the self-encoder using the feature representation P. Because they are more accurate to the original input, the features that were taken from the noisy samples are able to adapt to noisy situations. BL is simple and effective, as it processes raw data with minimal intervention. By applying LI approaches to prevailing minority class samples, our method improves minority class instance production. In contrast, network models offer a structured approach to managing and organizing data flow. Networks are thoughtfully modelled to outperform those that are merely assembled, as they consider the specific needs of the network. A well-designed network enhances robustness and overall performance. Our research proposes the ABL model to enhance BL's ability to extract characteristics from complex and fragmented medical data.

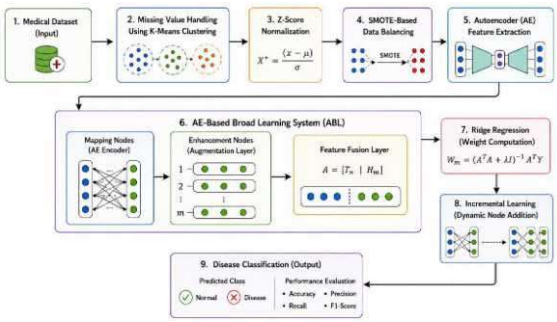

Figure 1. Proposed flowchart for disease classification using data preprocessing, feature extraction, broad learning, and classification.

It does this by incorporating AE into BL's architectural design. The proposed model ensures the model's effective computational capability and enhances its FE capabilities by combining AE and BL elements. Forward propagation and reverse propagation are the two main stages of DNN training. As data passes from the input layer to the output layer during forward propagation, each neuron's output is calculated repeatedly across the layers. Back-propagation, on the other hand, initiates at the output layer and works back to the input layer, updating each NN neuron's weights in reverse order. Both the input layer and the mapping layer weights are randomly generated in the original BLS (Block Least Squares) model. This independent weight generation allows for versatility across various learning tasks. The loss function plays a crucial role in updating the weights layer by layer, with gradients being calculated for each layer to facilitate these updates. However, in DNNs with multiple layers, challenges such as the vanishing gradient problem and local optima can often arise. Gradient explosion, slow convergence, and other phenomena are noted. The training procedure for the proposed model is fairly similar to that of standard NN training when incremental learning is not utilized. The inability to alter certain parameters after they have been verified during NN development, such as the hidden layers, number of neurons, learning rate, and iterations, without first restarting the NN. It is essential to pre-design the network model, as any time new detection data is collected, the entire dataset will need to be retrained. Setting the sample size and network parameters is the first step in training. One such example is shown in Fig 1. The two primary components of the model's hidden layer are improvement nodes and mapping nodes. The feature T_n specifies $[T_1, T_2, \ldots, T_n]$ is implemented by AE and the encoder mapping layer. The features produced by the mapping layer are determined by augmentation layer. Here, the j^{th} node set is specified by H_j.

$$H_j = \xi_j \left(T^n W_{hj} + \beta_{hj}\right); j = 1,2,\ldots,m \qquad (2)$$

The non-linear activation function is ξ_j, biases β_{hj} and random weights W_{hj}. Subsequently, m sets of nodes are integrated to give $H_m = [H_1 \cdot H_2, \cdots, H_m]$. The outputs from the mapping layer and the augmented layer are combined to provide the final output is represented as $A = [T^n | H^m]$.

$$Y = AW^m \qquad (3)$$

In this instance, W_m is the weight that the hidden layer sends to the output layer. Since the mapping layer, the auto-encoder (AE), contains fixed parameters during training, learning the Wm weights is the main objective for the entire network.

Additionally, because β_{hj} and W_m in the enhancement layer are generated at random and remain constant, ridge regression facilitates the determination of W_m. In contrast, traditional NNs utilize a method known as gradient descent to gradually update their weights. The weights in the width learning system are determined all at once; there is no weight update process. In contrast to DNNs, the model is simpler to model with lesser updating parameters and evaluates weights more rapidly. The mapping layer of the model contains a structure, which improves its FE capabilities. Furthermore, the architecture presented in this study exhibits dynamic properties throughout the training process, which sets it apart from existing DNN models. This adaptability makes it possible for effective updates through incremental learning, which raises the model's training efficiency. The model's ability to apply incremental learning for fast reconstruction ensures performance and the AE's role as the BLS mapping layer for FE improves the model's applicability for complex data conditions. By continuously expanding the model's structure, the dynamic incremental learning technique allows for quick and simple updates to the model's weights. This method not only enhances the model's capabilities but also optimally leverages the knowledge gained from the previously trained model. The training strategy for the proposed model is presented. Adding additional improved nodes is an effective way to boost the model's performance when training is unable to offer the required accuracy.

IV. NUMERICAL RESULTS AND DISCUSSION

The dataset's attacks are grouped together as abnormal and given the label 2, whereas the binary classification experiment's normal data is given the label 1. The classification experiment is evaluated using a number of metrics, including the ID precision rate (P), accuracy rate (Acc), recall, FPR, F-score, TPR, and others. Fig 2 – Fig 4 displays the indication description, and the computation. Metrics like: accuracy, precision, sensitivity and specificity are evaluated.

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

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

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

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

The *TP,FP,TN,and FN* metrics represent the trade-off to categorization mistakes. Usually, the ROC curve is depicted as FP rate vs. TR frequency. The work considers BoT-IoT, IoT network, MQTT, IoT-23, IoT-DS-1 and IoT-DS-2 respectively.

Table 1 Comparison with iteration 1

Dataset	EDE			
	Acc	Pre	Re	F1-score
1	99.1	100	99.8	99.2
2	97.2	98	97.5	97.2
3	99.3	99.1	99.5	99
4	99	99.6	99.6	99.6
5	98.9	99.2	98	98.5
6	99.9	99.6	99	99

Table 2 Comparison with iteration 2

Dataset	EDE			
	Acc	Pre	Re	F1-score
1	99.5	99.5	99	99.9
2	98	98	97	97.5
3	99.5	99.9	99	99.9
4	99.9	99.3	99.7	99.9
5	99.5	99.1	98.5	98.1
6	99.5	99.5	99.6	99.5

Table 3 Comparison with iteration 3

Dataset	EDE			
	Acc	Pre	Re	F1-score
1	99.5	99.80	99.83	99.8
2	97.7	97.90	97.10	97.10
3	99.91	99.90	99.1	99.80
4	99.90	99.75	99.76	99.75
5	99.40	99.20	98.20	98.20
6	99.10	99.10	99.10	99.10

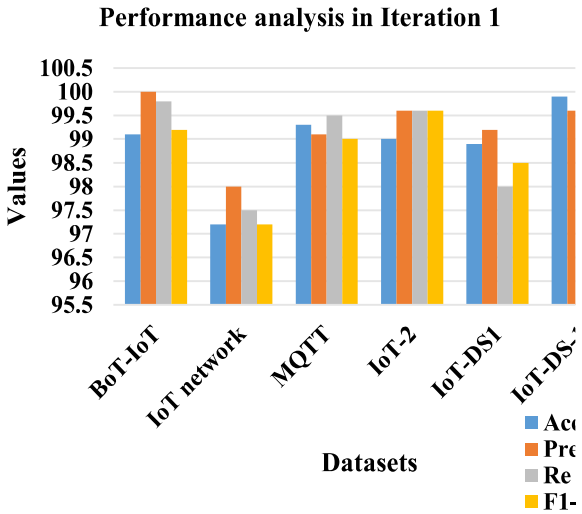

Fig 2 Performance evaluation with Iteration 1

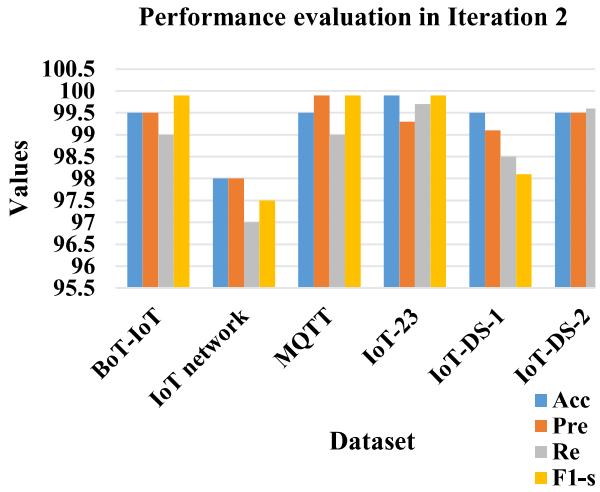

Fig 3 Performance evaluation with Iteration 2

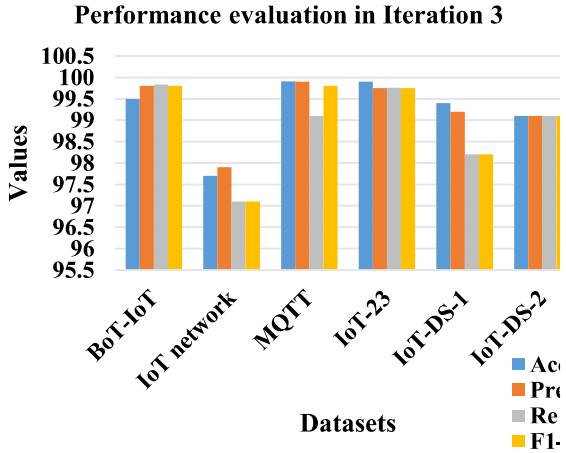

Fig 4 Evaluation with Iteration 3

Fig 3 to Fig 5 represent the evaluation of different approaches dataset. The performance is superior in superior to other methods. The adversaries are monitored and the attack traces are identified with the anticipated model.

V. CONCLUSION

On the basis of an enhanced data encoder (EDE), we suggest an ID technique. The proposed method is used to develop a new EDE classification model. This model efficiently extracts features from intricate high-dimensional network intrusion data by utilizing EDE dimensionality reduction capabilities. We can greatly improve the EDE performance by integrating feature representation with it. The model utilizes EDE to ensure effective and its capabilities to quickly determine ideal parameters. In the end, this combination makes it easier for EDE to classify high-dimensional data. The results of the experiment demonstrate that: 1) the EDE model continuously does well in data categorization over a range of datasets; 2) the model effectively addresses the issues of low accuracy, precision, and TPRs related to existing approaches; and 3) it outperforms existing methods on network ID datasets in a number of evaluation metrics. In the future, we will broaden the model's application area and carry out more thorough studies on the dimensionality reduction and categorization.

REFERENCES

[1] Roberts, J., & Toft, J. (2019). Finding vulnerabilities in IoT devices: Ethical hacking of electronic locks (Dissertation). School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden.

[2] Aldweesh, A., Derhab, A., & Emam, A. Z. (2020). Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. Knowledge-Based Systems, 189, Article 105124.

[3] Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications, 50, 102419.

[4] Ge, M., Fu, X., Syed, N., Baig, Z., Teo, G., & Robles-Kelly, A. (2019). Deep learning-based intrusion detection for IoT networks. In Proceedings of the 24th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC) (pp. 256–265).

[5] Hassan, M. M., Gumaei, A., Alsanad, A., Alrubaian, M., & Fortino, G. (2020). A hybrid deep learning model for efficient intrusion detection in the big data environment. Information Sciences, 513, 386–396.

[6] Deng, L., Lee, M., & Wang, H. (2019). IoT data feature extraction and intrusion detection system for smart cities based on deep migration learning. International Journal of Information Management, 49, 533–545.

[7] Yin, C., Zhang, S., Wang, J., & Xiong, N. N. (2020). Anomaly detection based on recurrent convolutional autoencoder for IoT time series. IEEE Transactions on Systems, Man, and Cybernetics: Systems.

[8] Wang, X., Su, Y., Zhang, M., & Nie, J. (2020). A deep hierarchical network for packet-level malicious traffic detection. IEEE Access, 8, 201728–201740.

[9] Yang, Y., & Wang, F. (2019). Wireless network intrusion detection based on improved convolutional neural network. IEEE Access, 7, 64366–64374.

[10] Otomo, M., Liu, D., & Nayak, A. (2019). DL-IDS: A deep learning-based intrusion detection framework for securing IoT. Transactions on Emerging Telecommunications Technologies.

[11] Li, Y., Xu, Y., Liu, Z., Hou, H., Zheng, Y., Xin, Y., Zhao,

Y., & Cui, L. (2020). Robust detection for network intrusion of industrial IoT based on multi-CNN fusion. Measurement, 154, Article 107450.

[12] Kunang, Y. N., Nurmaini, S., Setiawan, D., & Suprapto, B.Y.(2021). Attack classification of an intrusion detection system using deep learning and hyperparameter optimization. Journal of Information Security and Applications, 58, Article 102804.

[13] Sherubha. (2020). Graph based event measurement for analyzing distributed anomalies in sensor networks. Sādhanā, 45(212). https://doi.org/10.1007/s12046-020-01451-w

[14] Sherubha. (2019). An efficient network threat detection and classification method using ANP-MVPS algorithm in wireless sensor networks. International Journal of Innovative Technology and Exploring Engineering (IJITEE), 8(11).

[15] Sherubha. (2019). An efficient intrusion detection and authentication mechanism for detecting clone attack in wireless sensor networks. Journal of Advanced Research in Dynamical and Control Systems, 11(5), 55–68.