Journal of Information Systems Engineering and Management
2025, 10(26s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Adaptive Cyber Defense Through Deep Learning Technologies

[1]Smitha G V, [2]Dr. Samitha Khaiyum, [3]Gagan K H, [4]Pallavi S, [5]Omkar M Maradi, [6]NishantH N V,
[1]Assistant professor Department of MCA Dayananda Sagar College of Engineering
[2]Professor Department of MCA Dayananda Sagar College of Engineering
[3]Department of MCA Dayananda Sagar College of Engineering
[4]Department of MCA Dayananda Sagar College of Engineering
[5]Department of MCA Dayananda Sagar College of Engineering
[6]Department of MCA Dayananda Sagar College of Engineering

ARTICLE INFO

Received: 04 Jan 2025

Revised: 26 Feb 2025

Accepted: 06 Mar 2025

ABSTRACT

Introduction: The rapid adoption of cloud has led to a rise in cyber-attacks, increasing from 32 million in 2018 to 112 million in 2022, with a further 41% surge in early 2023. Traditional Intrusion Detection Systems (IDS) struggle with the complexity of cloud network threats due to the vast amount of network traffic and evolving attack types. Feature extraction is crucial for improving IDS accuracy, and Autoencoder (AE)-based deep learning methods offer a promising solution. This study explores the impact of AE feature learning on IDS performance, utilizing various autoencoder models like Variational, Sparse, Relational, and Denoising AEs. The system employs the NSL-KDD dataset and evaluates classifiers such as SVM, Random Forest, KNN, Gradient Boosting, and Logistic Regression to determine the most accurate model for detecting IoT threats.

Several studies highlight the importance of autonomous feature extraction in IDS to handle increasing cyber threats. Kunang Y N et al. achieved 86.96% accuracy using an Autoencoder and SVM on NSL-KDD, while Kushwaha P et al. improved attack detection through feature selection on KDD-CUP 99. Meng Q et al. introduced Relation Autoencoder for robust high-dimensional feature extraction, and Chae H S et al. optimized feature selection, reaching 99.794% accuracy with a decision tree. Yousefi-Azar M et al. proposed an AE-based dimensionality reduction method for efficient security. This paper advances previous work by comparing six machine-learning models on the latest NSL-KDD dataset.

The proposed model uses an autoencoder for feature extraction, improving network attack detection. The NSL-KDD dataset is split 80:20 for training and testing, with MinMaxScaler and one-hot encoding for preprocessing. A sparse autoencoder with dropout and regularization extracts key features while reducing data size. Six classifiers—K-NN, RF, GB, LR, DT, and SVM—analyze these features, each optimized to enhance accuracy and minimize overfitting. Performance metrics determine the best cyberattack detection method.

The proposed model uses an autoencoder-based deep learning approach for feature extraction, trained on the NSL_KDD dataset with normal traffic to address class imbalance. Data preprocessing includes MinMaxScaler and one-hot encoding, followed by feature extraction using a sparse autoencoder with dropout and regularization. Six classifiers (K-NN, RF, GB, SVM, DT, and LR) are evaluated, with K-NN achieving the highest accuracy of 99.28%. Performance metrics confirm the model's effectiveness in detecting network attacks.

The proposed system demonstrated that automatic feature extraction using an autoencoder significantly improves intrusion detection. By eliminating hand-crafted feature extraction and utilizing six ML classifiers, the system effectively classified cyberattacks from the NSL-KDD dataset. K-NN achieved the highest accuracy (99.28%), with RF also performing well. Performance metrics confirmed the model's effectiveness in enhancing network security. However, limitations include the use of an older dataset and limited experimentation. Future work will focus on larger datasets, improved speed, and AI-driven intrusion detection for cloud security.

Keywords: Cyber-attacks, Cloud Security, Intrusion Detection Systems (IDS), network data, traffic patterns, computational costs, auto encoder, Deep Learning Model, feature extraction,

machine learning classifiers, K-Nearest Neighbor (K-NN), Random Forest, Gradient Boosting, Logistic Regression, Decision Tree, Support Vector Machine (SVM), detection accuracy, and NSL_KDD dataset.

INTRODUCTION

The Cloud storage, which provides comprehensive connectivity and convenience, has become widely adopted. The risk of cyber-attacks has increased as a result, though, rising from 32 million in 2018 to 112 million in 2022 and then by 41% in the first two months of 2023. This equates to 60 attacks on 54% of firms every week on average. Malicious actors exploit Internet of Things devices by leveraging the weaknesses in how such devices become embedded in everyday life. Cyber-attack issues in the cloud networks scenario are analyzed in this introduction, which also brings into sharp focus the burgeoning complexity of threats and the imperative requirement for secure security controls to protect networked devices.

Good classification in the informatics engineering field, especially in information and network security, depends greatly on the extraction of meaningful properties. Among the greatest challenges is the enormous volume of network traffic data generated by cloud-based services and the expansion of the Internet of Things. Data complexity and the occurrence of new attack types affect intrusion detection systems' (IDS) performance. IDS employs techniques such as Decision trees [1], Back Propagation Neural Networks [2], K-means [3], and SVM [4], but one of the limitations is that there is still no automatic feature extraction.In many areas, feature extraction is necessary to significantly enhance detection and segmentation Digital images [5], text document categorization [6], speech recognition [8], audio and language identification [7], and biological signal processing [9][10] are all affected. Laplacian Eigenmaps, ISOMAP, Multidimensional Scaling, and Locally Linear Embedding all fail to handle large datasets. To make an IDS algorithm more accurate, unstructured input needs to be converted into applicable features. As part of learning for effective attack detection, this process includes feature extraction.

This work investigates the impact of Autoencoder-based feature learning on an IDS and how varying Autoencoder hyperparameters can increase the attack detection accuracy. SVM is the major evaluation metric utilized [11].In order to address these issues, a deep learning method uses Auto encoder (AE) to solve the feature extraction problem. By stacking layers and decreasing the reconstruction layer, AE employs a hidden layer to decrease dimensionality. AE's capacity to extract significant features is demonstrated by a variety of auto encoder variants, including Variational Autoencoder [12], Sparse Autoencoder, Relational Autoencoder [13], and Denoising Autoencoder [14]. Autoencoders are perfect for managing high-dimensional information because they can effectively learn hierarchical data representations without human assistance. In contrast to possible scalability challenges with feature extraction done manually, they are strong enough to handle enormous and complicated datasets. AEs are extremely efficient at identifying complicated correlatio ns and structures, like nonlinearities, which are difficult for human approaches. They automatically learn how to adjust to the data structure.

The novelty of the system lies in an implementation of feature extraction of an AE deep neural network model and the use of various machine\learning classifiers for detecting IoT network threats. With the usage of multiple machine-learning algorithms, the optimal model is suggested within accuracy constraints. This suggested system uses the NSL_KDD dataset, which is well-known for assessing intrusion detection techniques, to create a sparse autoencoder with dropout.When trained solely on "Normal" samples, the model minimizes mean squared error, guaranteeing efficient learning of typical behaviors. A denoising autoencoder is created using regularization and dropout technique to avoid over-fitting.The model is the trained by the system across ten epochs. Classifiers like Random Forest (RF), K-Nearest Neighbor (KNN), Gradient Boosting (GB), Logistic Regression (LG), Decision Tree (DT), and Support Vector Machine (SVM) are employed for the AE features. Classifier performance is evaluated using performance measures.

OBJECTIVES

Kunang Y N et al. conducted a research on the application of autonomous feature extraction in IDS. Due to the problem of increased network traffic and new forms of attacks, the research highlights the need for autonomous feature extraction.Employing a Support Vector Machine and Autoencoder, the paper model is tested on NSL_KDD and provides 86.96% accuracy.The study authors experimented thoroughly, empl oying GPU-accelerated

TensorFlow, and hyperparameter- tuned to enhance the result's stability. The study demonstrates the effectiveness of autonomous feature extraction in converting and classifying incursion data and demonstrates its superiority against traditional methods [1]. Kushwaha P et al. addressed important network security issues by concentrating on problems like phishing, probing, Denial of Service (DoS), and website defacements. To improve IDS effectiveness, the authors provide an algorithm that separates abnormal from typical connections. The study found that feature selection significantly improves model performance through testing on the KDD-CUP 99 dataset. The study uses statistical methods to classify attacks more quickly and with higher accuracy rates. Also, it identifies the challenges brought about by noise in intrusion detection and highlights how important robust feature selection is to reducing false alarms [2]. Meng Q et al. proposed a novel feature extraction method for high- dimensional data features known as Relation Autoencoder. The new method takes into consideration data dimensions as well as data relations. Feature extraction using data relations produces robust feat ures superior to any other autoencoder model. Data relationships have been stated to be of extremely great signi ficance to carry out correct dimensionality reduction as explained in work [3].

Chae H S et al. thought about intrusion detection and feature selection in the optimal sense in relation to the problem of traffic congestion in the network. Information Gain, Gain Ratio, and CFS were used for feature selection in the research and a new approach with attribute average of all as well as each individual classes of data being taken into account was developed. On NSL_KDD dataset, the existing method was superior to other methods with an accuracy rate of 99.794% based on a decision tree classifier on only 22 features. With up to 22 features in the research, accuracy and attribute ranking were found to be inversely related [4].

A novel feature learning approach was presented by Yousefi-Azar Metal. An AE is used by the model to learn latent representations. The AE reduces dimensionality and provides discriminative features by capturing semantic similarity among input features. The main contributions include dimensionality reduction for realistic implementation in small devices, low feature utilization, the efficacy of a single model for a variety of security tasks, and an unsupervised feature learning approach employing AE. Accuracy, multi-class logarithmic loss, and confusion matrices are examples of evaluation metrics [5]. There were fewer algorithms or an old dataset (KDD_CUP_99) used in the existing work. Conversely, the current paper compares six machine- learning techniques with the most recent dataset (NSL_KDD).

METHODS

A. Proposed Method

The proposed model applies an autoencoder deep learning model for the purpose of feature extraction with a focus on improved network attack detection. The use of this approach surpasses other more traditional approaches to feature extraction. TRAIN_KDD and TEST_KDD were obtained by partitioning the NSL_KDD dataset in the proportion 80:20 in an effort to train and test the proposed model. 43 columns make up the dataset, but user-defined column names are assigned because there are no established column names. 37 attack types are included in the outcome column and are divided into four groups. Normal, DOS, Probe, Privilege, and Access assaults are among the user-defined list of attacks that is generated. The dataset's outcome column is swapped out for a new 'Class' column in the suggested model. This column specifies the relevant attack from the user-defined list and indicates the attack category.

The suggested system uses two methods to prepare features. The first method, called "MinMaxScaler," helps algorithms that need a bounded input space by transforming data into a predetermined range. Onehot encoding is the second, which transforms discrete data like "protocol_type," "service," and "flag" into numerical values. Following preprocessing, the AE deep learning model extracts features from the TRAIN_KDD and TEST_KDD datasets. For cyberattack prediction, the system compares classifiers such as K-NN, Gradient Boosting, Logistic Regression, Random Forest, SVM, and Decision Tree. Performance metrics scores are utilized to compare the performance of every classifier. The system architecture of the proposed framework is depicted by the way the proposed system extensively compare and picks the best technologies to identify the best cyberattack detection methods.

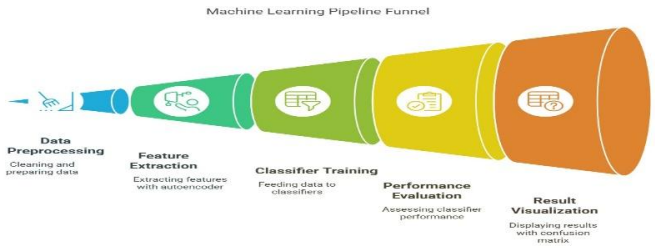

Fig.1 System design of the proposed work

B. Autoencoder

The system in question utilizes an AE deep learning model to extract features. An AE is a form of neural network that makes unsupervised learning and dimensionality reduct ion easier. This is achieved through its encoder-decoder structure. The decoder rebuilds the data that was originally inputted after the encoder compresses the input data. Reducing the size of the input data with no loss to its key features is the core aim of AE. Anomaly detection and feature extraction are augmented by the decoder reconstructing the input data and the encoder compressing the input data. The method can be applied in a range of different applications, such as signal and image processing.

AEs can be applied to data denoising and generation tasks because they are capable of learning meaningful representations without requiring labeled data. They are universal and applicable in a broad variety of applications due to their straightforward architecture, which enables efficient data compression. The AE deep learning model architecture diagram and process flow are shown in Fig. 2. [15]. The architecture diagram and process flow are as follows:

A sparse autoencoder with input dropout is implemented in the suggested system. Eight neurons make up the hidden layer of the model, whereas 122 neurons make up the input layer. Because of the resulting compression ratio of 122/8, the AE is compelled to identify trends and connections among the characteristics. There are 122 ReLU activation units in the output layer. "Normal" samples are utilized to train the AE to learn the identity function and minimize mean squared err or. Input dropout makes it a denoising autoencoder, and regularization is applied to prevent overfitting. Ten epochs of size 100 are utilized for training, and 10% of normal samples are utilized for validation.

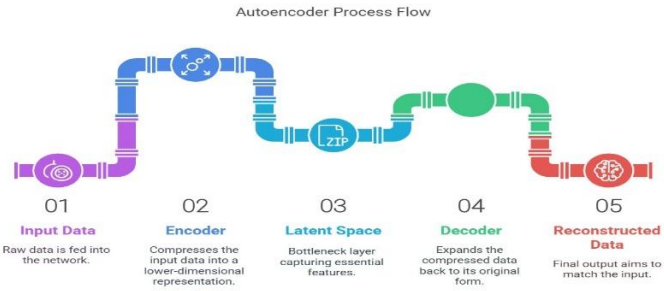

Fig. 2. Autoencoder Architectural Diagram

C. *Classifiers*

K-NN, RF, GB, LG, DT, and SVM are some of the six machine learning classifiers utilized by this system.They are applied to forecast and study cyberattacks within the NSL_KDD dataset. An autoencoder deep learning model is utilized to extract the features employed in course of these forecasts. To generate predictions, every machine learning algorithm uses a particular set of parameters. High accuracy is the goal of the RF classifier while avoiding overfitting. It employs one hundred trees with a maximum depth of ten. Using five neighbors, K-NN increases sensitivity and detects outliers. LR iterates a thousand times in order to maximize the outcomes. To guarantee class separation, SVM employs a Radial Basis Function kernel. To identify patterns and avoid overfitting, GB combines 100 learners with a learning rate of 0.1 and a maximum depth of 3. DT is employed with a maximum depth of 5. The

testing procedure places a strong emphasis on choosing parameters carefully in order to detect cyberattacks as effectively as possible. Following testing, the suggested method assesses the performance of each ML classifier's metrics.

RESULTS

The suggested approach uses a deep learning model with an autoencoder to focus on feature extraction. GPU-enabled TensorFlow is used in the evaluation process, which takes place on a Windows 11 PC running Google Colab and equipped with a RYZEN 5-5500U processor, 16GB RAM, AMD-RADEON Graphics, and 512 GB SSD.

[1] Datasets

Data is gathered, preprocessed, and features are extracted using the NSL_KDD dataset [16]. It is a data source for offline network analysis based on the KDD'99 dataset and is an upgraded version of the KDD'99 dataset. This dataset, which has 43 attributes and one class attribute, is subjected to the suggested system. The NSL_KDD dataset is significantly smaller than the KDD'99 dataset and includes duplicate records. The procedure comprises loading and retrieving datasets with information on network traffic.

The 22 various kinds of attacks in the TRAIN_KDD dataset are utilized to train the model. The test phase is executed with the TEST_KDD dataset of 38 various kinds of attacks.As indicated in Table I, attacks are classified into four classes during training and testing.

Table I. Attacks are categorized for the test and train datasets.

TRAIN_DATASET		**TEST_DATASET**	
Attack_Type	**Occurrences**	**Attack_Type**	**Occurrences**
NORMAL	67352	NORMAL	9855
DOS	45927	DOS	7459
PROBE	11656	PROBE	2421
PRIVILEGE	43	PRIVILEGE	65
ACCESS	995	ACCESS	2743
TOTAL	**125973**	**TOTAL**	**22543**

The frequency of attacks for each Attack_Type for the NSL_KDD dataset is shown in Table II. There are 22,543 occurrences in the testing set and 125,973 instances in the training set.

Table II**.** Count of incidents for each type of attack

The KDD_99 and NSL_KDD datasets have problems with class imbalance and unrealistic data, which the suggested solution seeks to remedy. To alleviate these problems, the system was trained on normal traffic alone, without the use of attack data. This avoids the class imbalance of the dataset from impacting the model. Second, the system is more convenient to utilize. In real- world use and better suited to apply in real networks because it utilizes only normal traffic data in training.

[2] *Performance Measures*

DOS	**PROBE**	**PRIVILEGE**	**ACCESS**
processtable smurf teardrop land mailbomb pod worm back udpstorm Neptune apache2	nmap saint mscan portsweep satan ipsweep	sqlattack loadmodule rootkit perl ps xterm buffer_overflow	warezclient phf sendmail ftp_write imap snoop snmpgetattack http_tunnel snmpguessspy xclock named warezmaster guess_passwd

A classification technique produces its output in the form of a confusion matrix.The matrix helps in calculating the

accuracy of the model by predicting the correct or incorrect labels.Precision, recall accuracy, and F1-score can be derived to measure the reliability and efficacy of the model.

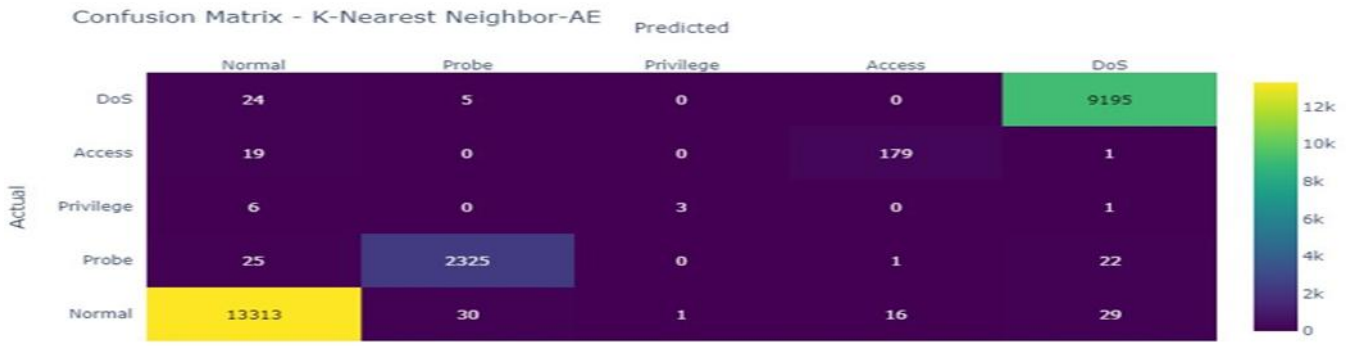

Fig. 3.Matrix of K-Nearest Neighbor-AE Confusion

A confusion matrix among the performances of different machine learning classifiers is presented in the material. The most accurate of all the classifiers is the K-Nearest Neighbor classifier, as the matrix clearly shows. In a confusion matrix, the columns represent the expected classes, and the rows represent the actual classes. The K-NN confusion matrix is depicted in Fig. 3. It can be seen that the K-NN classifier is not good at identifying Privilege and Probe attacks, but it is extremely good at identifying DoS and Access attacks. To effectively detect network attacks on larger data sets, features are typically extracted by an autoencoder deep learning model and then used with different machine learning methods. For machine learning classifiers, the most accurate one is K-NN. Below is a list of the formulas for performance metrics:

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (1)$$

$$Precision = \frac{(TP)}{(TP+FP)} \quad (2)$$

$$Recall = \frac{(TP)}{(TP+FN)} \quad (3)$$

$$F1 - Score = \frac{2 \text{ x Precision x Recall}}{\text{Precision+Recall}} \quad (4)$$

[3] Experimental Results

The NSL_KDD dataset was collected, and the training and testing datasets column names were standardized. Both datasets should have the outcome column deleted in order to get the data ready for analysis. The class of assault in each row must then be specified by updating the "Class" column based on the attack types. One-hot encoding must then be used on the discrete features after scaling the values using MinMaxScaler. The suggested system employed a sparse autoencoder model with a 122/8 compression ratio following preprocessing. It included input and dropout layers, an 8- neuron hidden layer, and a 122-unit ReLU-activated output layer. It uses input dropout for denoising, regularization to avoid overfitting, and mean squared error minimization after being trained on "Normal" data. Robust learning and pattern extraction in the AE are ensured by 10% validation on normal samples after the model is trained for 10 epochs using an Adam optimizer with a batch size set to 100. K-NN, RF, GB, LG, DT, and SVM were among the several ML classifiers used to assess the feature extraction outcomes. Each classifier's parameters were chosen appropriately so that predictions could be made.

Model	Accuracy	Recall	Precision	F1 Score
Random Forest-AE	0.990634	0.990634	0.990534	0.990448
K-Nearest Neighbor- AE	0.992856	0.992856	0.992784	0.992778
Logistic Regression- AE	0.943678	0.943678	0.937825	0.940414

Support Vector Machine-AE	0.976465	0.976465	0.975934	0.975548
Gradient Boosting- AE	0.966582	0.966582	0.978075	0.972088
Decision Tree-AE	0.949594	0.949594	0.943521	0.946191

Fig. 4. Measures of performance for different machine learning classifiers.

K-NN had the best accuracy in the evaluation of various ML classifiers, attaining an astounding 99.28%. With a 99% accuracy rate, RF performed admirably in close pursuit. Additionally, SVM and GB produced noteworthy accuracy rates of 97.58% and 98.31%, respectively. The impressive accuracy of 94.35% was attained via DT. However, with an accuracy of 92.81%, LR had the lowest performance among the machine learning classifiers. This comparison study highlights the differences in performance between several machine-learning techniques while highlighting the efficacy of K-NN and RF.The performance metrics for each classifier are shown in the table plot in Figure 4.

DISCUSSION

In conclusion, the system in question proved that automatic feature extraction could be utilized to improve intrusion detection systems. Precisely, the system utilized an Autoencoder deep learning algorithm that effectively and unbiasedly determined attack types from a wide variety of different types utilizing the NSL KDD dataset and excluding hand-crafted feature extraction techniques. The system subsequently proceeded to utilize six Machine Learning (ML) classifiers, such as K-Nearest Neighbor (K- NN), Random Forest (RF), Gradient Boosting, Logistic Regression, Decision Tree, and Support Vector Machine, to classify cyberattacks on the basis of features produced by the Autoencoder model. With balanced precision and recall scores, the K-NN and RF models demonstrated remarkable accuracy. F1-score, accuracy, recall, and precision were among the performance indicators assessed. While the Logistic Regression model produced dependable performance despite a slightly lower accuracy rate, the K-NN method earned the maximum accuracy at 99.28%. The confusion matrices offered insightful information about each machine learning algorithm's capacity for prediction. All things considered, this suggested solution demonstrated how automated feature extraction improved network security procedures and aided in the continuous advancement of intrusion detection systems.This study has some limitations as it was done based on an old dataset and less experimentation. Future studies are likely to utilize newer and larger data for feature extraction. Speed will be explored in future studies, but this was not the main subject of this study. Creating a robust intrusion detection system for the Internet of Things driven by artificial intelligence and on state-of-the-art machine learning algorithms is also the aim.

REFRENCES

[1] Hee-su Chae and Sang hyun Choi, “Feature Selection for efficient Intrusion Detection using attribute ratio,” vol. 8, 2014.

[2] Bhavin Shah and Bhushan H. Trivedi, “Reducing Features of KDD CUP 1999 Dataset for Anomaly Detection Using Back Propagation Neural Network,” 2015.

[3] Deeman Yousif Mahmood and Mohammed Abdullah Hussein,“Feature-Based Unsupervised Intrusion Detection,” vol. 8, 2014.

[4] Prashant Kushwaha, H. Buckchash, and Balasubramanian Raman, “Anomaly-based intrusion detection using filter- based feature selection on KDD-CUP 99,” in Region 10 Conference, TENCON 2017-2017 IEEE.

[5] Yuhandri, S. Madenda, E. P. Wibowo, and - Karmilasari, “Object Feature Extraction of Songket Image Using Chain Code Algorithm,” International Journal on Advanced Science, Engineering and Information Technology, vol. 7, Feb. 2017.

[6] Charles Henrique Porto Ferreira, D. M. R. de Medeiros, and Fabricio Olivetti de França, “DCDistance: A Supervised Text Document Feature extraction based on class labels,” Jan. 2018.

[7] Fred Richardson, Douglas Reynolds, and Najim Dehak, “Deep Neural Network Approaches to Speaker and Language Recognition,” IEEE Signal Processing Letters, vol. 22, Oct. 2015.

[8] J Barker, R Marxer, E Vincent, and Shinji Watanabe, “The third ‘CHiME’ speech separation and recognition challenge: Dataset, task and baselines,” 2015.

[9] D Sinha, K Sasirekha, K Thangavel, ”GPU based epileptic seizure detection using deep autoencoder with particle swarm optimization,” In Journal of Physics: Conference Series (Vol.2318, No. 1, p. 012010). IOP Publishing.
[10] R Jenke, A Peer, and M Buss, “Feature Extraction and Selection for Emotion Recognition from EEG,” IEEE Transactions on Affective Computing, vol. 5, no. 3, pp. 327– 339, Jul. 2014.
[11] YN Kunang ,S Nurmaini,D Stiawan, and A Zarkasi, “Automatic features extraction using autoencoder in intrusion detection system,” Oct 2018.
[12] Jinwon An and Sungzoon Cho, “Variational Autoencoder Based Anomaly Detection Using Reconstruction Probability,” Technical Report,2015.
[13] Q Meng, D Catchpoole, D Skillicorn, and Paul J. Kennedy, “Relational Autoencoder for Feature Extraction,” May 2017.
[14] Y Wang, W Cai, and P Wei, “A Deep Learning Approach for Detecting Malicious Javascript Code,” Security Comm. Networks, vol. 9, Jul. 2016.
[15] S Aggarwal, (2023) , ”Research on Anomaly Detection in Time Series: Exploring United States Exports and Imports Using Long Short-Term Memory,” Journal of Research, Innovation and Technologies.
[16] https://www.kaggle.com/datasets/hassan06/nslkdd