Hybrid Variational Autoencoders-DenseNet Convolution Neural-Network Based Deep Learning Approach for Intrusion Detection System

Om Prakash Panjiyar[1,*], Babu R. Dawadi[2]
[1]Acme Engineering College, Purbanchal University (PU), Kathmandu, Nepal
[2]Institute of Engineering, Tribhuvan University (TU), Lalitpur, Nepal
*Corresponding author: omprakashpanjiyar@gmail.com

Abstract
In the modern big data environment, a deep learning approach for intrusion detection was seen as more effective and efficient to counter known as well as unknown attacks. Much research has been conducted on intrusion detection systems using a deep learning approach, as well as a hybrid model of deep learning, and continues to be conducted to minimize the risk of intrusion. In this paper hybrid approach is used using variational autoencoders and DenseNet convolutional neural networks to model an intrusion detection system. An experiment is performed using the NSL-KDD dataset to calculate the performance metrics of the model, which contains four types of attacks: probe attack, DoS attack, user-to-root attack, and remote to local attack. The result obtained is compared with the results of other related works. The result obtained from the experiment is quite satisfactory as well, and in comparison with other related works, the model seems comparatively efficient. An intrusion detection system using a VAE-DCNN hybrid model can detect malicious activities very efficiently and correctly, as the performance metric of this model is quite good. The possibility of a false alarm will be reduced using this hybrid model, as the false alarm rate is low.

Keywords: Intrusion Detection System, Deep Learning, Variational Autoencoders, DenseNet Convolutional Neural Network, VAE-DCNN, NSL-KDD

1. Introduction

In today's world of technology, information is stored, analyzed, exchanged between two parties, manipulated, and accessed by authorized users; all are done using advanced technology. Business enterprises, small or big, government bodies are moving to advance technology for storing their critical data, to analyze the data and extract necessary information from the raw data, to exchange necessary information with authorized users to operate daily business operations. As the technology is getting more advanced, malicious activities to tamper with critical data and information are also getting more sophisticated, and a great challenge to identify unknown and unpredictable malware, which are very complicated to identify with traditional intrusion detection systems like firewalls. Intruder uses many malicious codes and viruses to get access to computing systems, storage systems, information systems, internally within premises infrastructure, or externally via internet access, gaining access to internal IT infrastructure. There are many malicious attacks and viruses which are well known to IT industry with known patterns such as DDOS attack, DOS attack, Trojan horse, Phishing etc., which can be easily detected and prevented by existing intrusion detection system and prevention detection system but computer attacks and viruses used by intruders are getting more sophisticated that cannot be identified easily with normal intrusion detection system, for that more intelligent intrusion detection are required to develop. Security threats such as zero-day attacks, computer software vulnerabilities that were not known before are increasing nowadays, and are very complicated issues regarding cybersecurity. Attackers find vulnerabilities within software used in computing systems, storage systems, and networking, and misuse these vulnerabilities to gain access to the systems and cause illegal activities. The Symantec Internet Security Threat Report for 2017 shows that in 2016, zero-day attacks were more than three billion higher than previously recorded [22]. According to the 2017 Data Breach Statistics, hackers had stolen or compromised nearly nine billion data records since 2013 [3]. To ensure the Confidentiality, Integrity, and Availability (CIA) triad of information security systems, proper intrusion detection systems should be designed and developed, which are beyond the capability of traditional firewalls [11]. Threats come from both internally and externally via the internet, so advanced host-based intrusion systems to check the internal attack need to be designed, and advanced network-based intrusion systems to check external attacks coming via the internet need to be designed.

In the last few decades, a deep learning approach has been used in intrusion detection systems to improve the capability of intrusion detection systems so that known and unknown intrusions can be detected and prevented efficiently. Much research has been done to make intrusion detection systems more intelligent using many machine learning algorithms, and still, research is going on by different researchers at different places around the globe. Supervised, unsupervised, and semi-supervised learning algorithms are used to make a model of an intrusion detection system so that known as well as unknown threats can be easily and efficiently detected, and necessary steps can be taken to minimize them.

Article History:
Submission: 2025-05-05, Revision: 2025-09-24, Acceptance: 2025-11-17

In my thesis, I am going to design a model for an intrusion detection system using well well-known supervised deep learning algorithm, DenseNet convolutional neural network, and an unsupervised deep learning algorithm, variational autoencoder neural network. A hybrid Variational autoencoder-DenseNet convolutional neural network model will be designed, making intrusion detection systems intelligent to detect known as well as unknown threats and attacks.

A. Various computer attack types addressed in this proposed model are [21]

1. Probing Attack

Probing attack means gathering all information about the design structure and source codes of the networking devices and computer systems. After gathering necessary information, attackers do reverse engineering of the system or find existing vulnerabilities and prepare for an attack.

2. Denial of Service (DOS) Attack

In denial of service (DOS) attacks, attackers flood the target system with unnecessary traffic, making the system busy and making it inaccessible to authorized intended users. The heavy, unnecessary network traffic system is either shut down or made inaccessible to the intended legal users.

3. User-to-Root (U2R) Attack

User-to-root (U2R) attack is a type of attack in which intruders gain access to a network or computer system as valid legal users. In this attack, intruders access the IT infrastructure as authorized users and tamper with the necessary critical data, and engage in illegal activities.

4. Remote-to-Local (R2L) Attack

Remote-to-Local (R2L) attacks occur when packets are sent to a target machine, allowing the attacker to monitor user activities and eventually acquire privileges similar to those of a standard end user on the system.

B. Objective of Study

1. Use a hybrid Variational autoencoders-DenseNet convolutional neural network to model an intrusion detection system.

2. Improve the efficiency of the detection of malicious activities on the system.

3. Reduce the possibility of false alarms.

2. Literature Review

Today's world is the world of technology, all private as well as government organizations are moving towards the use of ICT infrastructure for their day to day operations and fully depend on internet, database and ICT computing infrastructure to share files and data among employee as well as among different partners, stores vital data and to compute complex problem ("Information and Communication Technology," n.d.). As the demand for ICT is increasing day by day, threats to the confidentiality, integrity, and availability of ICT infrastructure are increasing day by day [22]. Traditionally, firewalls were used to detect malicious activities, which are not sufficient measures to check the attack. Gradually, intrusion detection systems are being invented to detect known attacks based on known patterns of attack, but zero-day attacks are increasing rapidly (Breach Level Index, 2017). To detect zero-day attacks, researchers are researching intrusion detection systems based on machine learning and deep learning. Deep learning method is an effective method to model an intrusion detection system to detect known as well as unknown malicious activities on a system [21]. Many deep learning algorithms, as well as hybrid deep learning algorithms, can be used to model intrusion detection systems, among which variational autoencoders and convolutional neural networks are being used in this to make a hybrid model to detect known as well as unknown attacks.

A. Intrusion Detection System

Intrusion in cybersecurity is any kind of unauthorized access to information systems, which will hamper the Confidentiality, Integrity, Availability (CIA) triad of information security systems. For example, DOS attacks make systems inaccessible to intended users, violating the availability of information systems. IDS can be implemented as either a software application or a hardware device, which has the capability to detect malicious activities in network traffic or internal host computer systems [11], [24]. The goal of an intrusion detection system is to monitor network traffic and host system logs to detect malicious activities in network traffic as well as in computer resources. An intrusion detection system plays a vital role in the protection of the CIA triad of information security systems, which traditional firewalls can't do [17].

B. Classification of Intrusion Detection Systems

Intruders can attack directly to the host system directly or via network traffic, to hamper the server, host system, or workstation alone or by making high unnecessary network traffic to prevent the access of the intended user. According to [16] and as shown in Figure 1 below, the protected intrusion detection system is divided into three distinct categories.

Based on deployment location, intrusion detection system is classified into three types. Deployment location can be at a particular host, such as a computer, desktop, server, or workstation, or at the gate of the ICT infrastructure network to monitor all incoming traffic. Based on deployment location, IDS can be divided into host-based, network-based, and hybrid-based systems, as detailed in the following section [5], [6], [15].

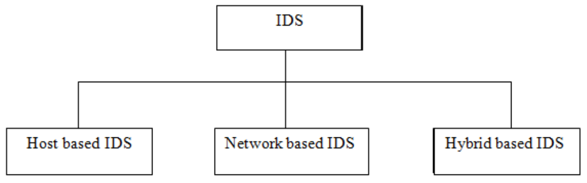

Figure 1. Classification of IDS on the basis of Protected System [16].

a. Host-Based IDS

This type of intrusion detection system is placed in a single device such as a personal computer, data server, workstation, web server, etc., which monitors the activities of a single device and detects any malicious activities that occur. HIDS is limited to only the device where that HIDS is installed; it cannot monitor the whole network. HIDS monitor the operating system of the host and all its log files and trigger an alarm if any malicious activities are found [5].

b. Network-Based IDS

Network-based IDS monitors' whole traffic entering the network of the IT infrastructure; it is placed along network segments or at the boundaries of the network. NIDS monitors the whole incoming traffic and analyzes it if any malicious traffic is found; it triggers an alarm to notify [6].

c. Hybrid-Based IDS

To mitigate the drawback of the above-mentioned detection system hybrid-based IDS is used, in which both HIDS and NIDS are used to increase the efficiency of detection of malicious attacks [15].

C. Intrusion Detection System Techniques

Some malicious attacks are well known to the information security community, whose patterns are familiar, whereas some malicious attacks are not known and have not appeared before, like zero-day attacks. On the basis of this, known and unknown malicious attack intrusion detection system is classified into two types, which are Signature-based intrusion detection systems (SIDS) and Anomaly-based intrusion detection systems (AIDS) as mentioned in Figure 2 below.

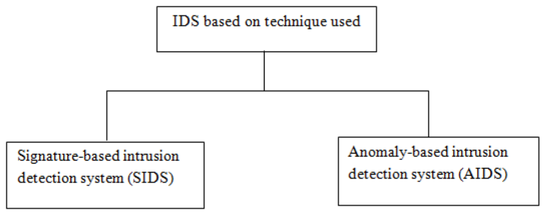
IDS based on technique used
Signature-based intrusion detection system (SIDS)
Anomaly-based intrusion detection system (AIDS)

Figure 2. Types of IDS based on Technique used [16]

According to the detection method, IDS is classified into two types, which are based on the pattern of malicious activities or behavior of the system, which are described below in the next section.

a. Signature-Based Intrusion Detection System (SIDS)

Signature-based intrusion detection system (SIDS) such a type of IDS that is based on pattern matching techniques to detect only known malicious attacks. In SIDS, all patterns of known attacks are stored in the signature database of the detection system, and all activities of the system are matched with the stored known signatures. If any activities of the system matched the known pattern, then IDS triggers an alarm indicating a malicious attack has occurred [14]. The main drawback of SIDS as it can only detect known threats. If any unknown threats appear on the system, it can detect them like a zero-day attack.

b. Anomaly-Based Intrusion Detection System (AIDS)

Anomaly-based intrusion detection system (AIDS) overcomes the limitation of SIDS. Normal behaviors of computer systems are modeled using machine learning, statistical-based, and knowledge-based techniques in AIDS. Behaviors of computer systems are monitored by AIDS, and if any deviation occurs on observed behavior or system with modeled behavior, then AIDS detects it as an anomaly and predicts it as a malicious attack. Unknown malicious attacks can be easily detected by AIDS, such as zero-day attacks. AIDS is a very efficient system to detect both known and unknown attacks on the information system. Nowadays, many machine learning algorithms are used to model the anomaly-based intrusion detection system (AIDS) [1], [19].

D. Variational Autoencoders (VAEs)

To understand the variational autoencoders, first, we have to understand the concept of simple autoencoders, which is an unsupervised two-stage neural network. The first stage of this network is the encoder, and the second stage is the decoder. Autoencoders consist of two neural networks, one each of encoder and decoder, as shown in Figure 3.

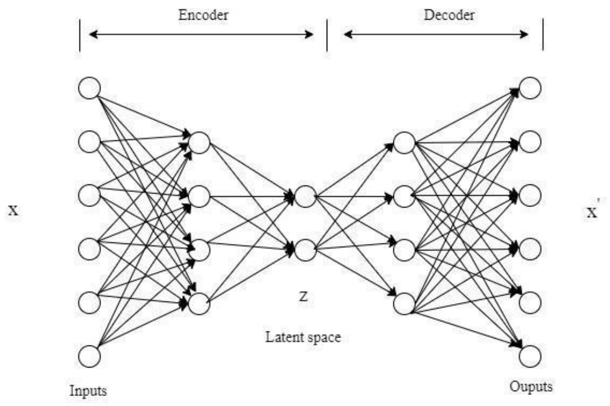

Figure 3. Fully Connected Autoencoders [10], [13]

The encoder takes x inputs and compresses them into a lower dimension, and passes them to the next layer. Final compression is done on bottleneck layers, and the lowest compressed dimensions are presented in the latent space, also called the latent vector z. Like AE, a VAE also consists of an encoder and a decoder. However, the key difference is that instead of learning a deterministic point in the latent space, VAE learns a distribution (mean and variance) for each input data point in the latent space. Next, the decoder takes the lower dimension latent space z as input and attempts to reconstruct the input, having the same number of hidden layers as well as the same number of neurons on the neural network of the encoder. The reconstructed output x' of the decoder is almost identical to the input x of the encoder. Reconstruction error occurs if there is a difference between reconstructed output x' and input x, which can be adjusted using an activation function [10], [13]; mathematically, it is formulated as r = $|x' - x|$.

This is the benefit of autoencoders to find anomalies in the system by fixing a threshold value and comparing it with the reconstruction error. If the reconstruction error is greater than threshold value, then it will be considered as an anomaly or it will be considered as normal traffic.

E. Convolutional Neural Network (CNN)

A convolutional neural network is one of the supervised (discriminative) types of deep learning techniques, specially modeled for image processing, and can be used efficiently and effectively for the classification of intrusion. As they can automatically learn from hierarchical feature representations from raw data, they require less processing. The internal connectivity between different layers of CNN resembles the human neural network. The CNN architecture contains three main layers: a convolutional layer, a pooling layer, and a fully connected layer, which are described below [20].

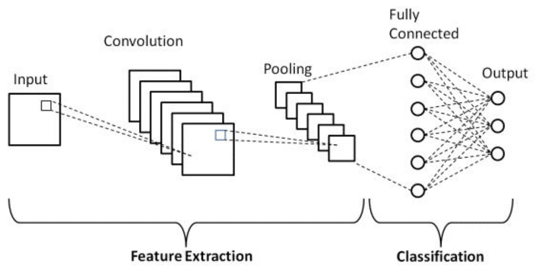

Figure 4. Fully Connected Autoencoders [10], [13]

a. Convolutional layer

The convolutional layer is a crucial component of a Convolutional Neural Network (CNN), responsible for performing computations to extract features from the input data. It utilizes a filter, also known as a feature detector, to scan the input data and identify key features. The filter is typically represented as a 2D array of weights, which corresponds to a portion of a 2D image. A 3×3 matrix filter is commonly used, though other filter sizes are also possible. To compute the dot product between the pixels, the filter is applied to a localized region of the input image and produces an output that is stored in an array. The filter then moves across the image, repeating the process until it has processed the entire dataset. The resulting output from these operations is referred to as the feature map. After each convolution, a ReLU (Rectified Linear Unit) activation function is typically applied to each feature map to introduce nonlinearity to the model. The pooling layer follows the convolutional layer, and together they form a convolutional block, which is essential for feature extraction and dimensionality reduction.

b. Pooling layer

Pooling layer is also called the down-sampling layer, reduces the dimensionality of the input. The pooling layer also uses a filter to sweep the whole input like a convolutional layer does, but it doesn't use weights. Pooling layer loses some information while reducing the dimensionality, causing a reduction of complexity and increasing efficiency. It also reduces the risk of overfitting. There are two types of pooling.

1. Average pooling

Average pooling calculates the receptive field's average value while scanning the input.

2. Max pooling

Max pooling sends the pixel with the maximum value to populate the output array.

c. Fully connected layer

A fully connected layer is the final layer of a convolutional network, which uses an activation function to classify the extracted features from the convolutional and pooling layers. Typically sigmoid activation function is used to classify the features. There are other options of activation functions also for classification, which can be used efficiently.

F. DenseNet Convolutional neural network (DCNN)

DenseNet is a type of convolutional neural network (CNN) where each layer is directly connected to every other layer in a feed-forward manner. In a traditional CNN with L layers, there are L connections, each linking a layer to its immediate successor. However, in a DenseNet, the number of connections between layers is given by the formula (L (L+1))/2 [5], [7], resulting in many more direct connections, as illustrated in Figure 5. DenseNets offer several significant advantages, including mitigating the vanishing-gradient problem, enhancing feature propagation, promoting the reuse of features, and dramatically reducing the overall number of parameters required for the network.

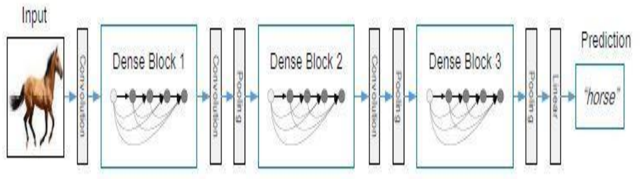

Figure 5. DenseNet Convolutional Neural Network (DCNN) [7]

G. NSL KDD Dataset

NSL KDD is an improved version of the KDD CUP'99 dataset to improve the efficiency of training and testing the intrusion detection models. KDD CUP '99 is deficient due to the presence of duplicate data, approximately 78 percentage in the train data and 75 percent in the test data, which makes the training and testing of the model less accurate. Due to this reason, a new edition of the network traffic dataset called NSL KDD is offered, which is a better version with no duplicate data than KDD CUP'99 [18].

Advantages of using the NSL KDD dataset over KDD CUP'99 are [18].

• NSL KDD Train+ data does not contain duplicate data, which makes the model an unbiased classifier.

• NSL KDD Test+ also contains no duplicate data, which makes trained models more efficient and make better decisions.

• The dataset contains a reasonable amount of sample data so that the whole train and test data can be used for training and testing of the model at once, without partitioning it into a small chunk of the dataset.

The NSL KDD dataset contains two sets of data, Train+ and Test+ datasets set which are used for training and testing of the model, respectively. The KDD Train+ data set is used to train the intrusion detection system model, which contains four classes of attacks and normal traffic data. The KDD Test+ data set is used to test the trained intrusion detection system to check the efficiency of the intrusion detection system, which also contains four classes of attacks and normal traffic data. Four classes of attacks are listed below:

1. DoS attack
2. Probe attack
3. Root to local (R2L) attack
4. User to root (U2R) attack

The percentage distribution of different attacks given above, along with normal data in NSL KDD Train+ data and NSL KDD Test+ data, is graphically presented below in a pie chart in Figures 6, 7.

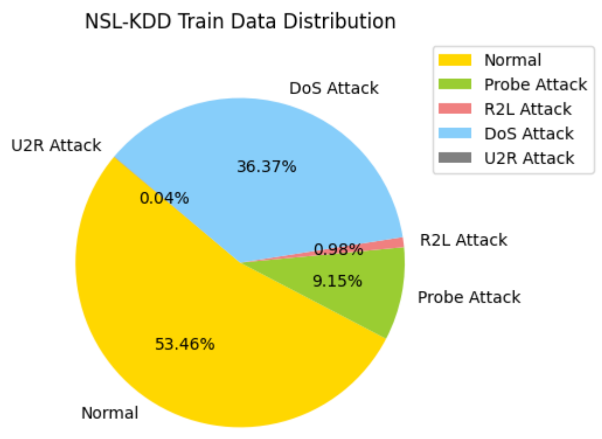

Figure 6. Distribution of NSL-KDD Train+ Data in Percentage

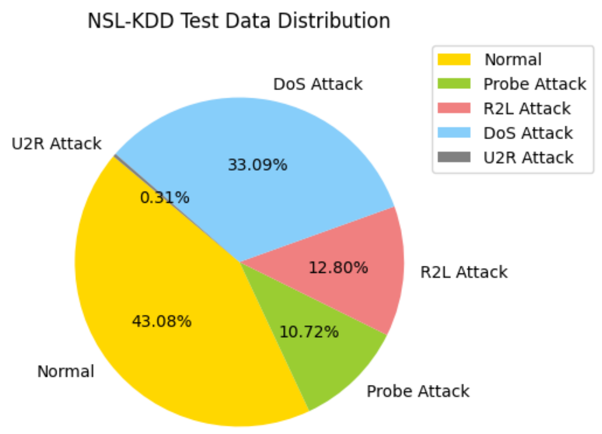

Figure 7. Distribution of NSL-KDD Test+ Data in Percentage

H. Trend analysis of related works

To analyze imported aspects, method, dataset, parameters, and concepts to formulate an efficient model for an intrusion detection system, various research works are studied, and trend analysis is done, which is summarized below.

a. Research Overview

• Performance Evaluation of NSL-KDD Dataset Using ANN [9]: This study evaluates the performance of an Artificial Neural Network (ANN) on the NSL-KDD dataset, focusing on the importance of feature selection to improve classification accuracy.

• Deep Learning Method Combining Sparse Autoencoder and SVM for Network Intrusion Detection [2]: This research introduces a deep learning method that combines Sparse Autoencoder with Support Vector Machine (SVM) to enhance intrusion detection, demonstrating significant improvements in accuracy.

• An Empirical Study on Multiclass Classification for Network Intrusion Detection [4]: This study investigates the use of multiclass classification in network intrusion detection, applying computational intelligence techniques and comparing various models for their practical effectiveness in real-world scenarios.

• A Novel Statistical and Autoencoder-Based Intelligent Intrusion Detection Approach [10](Ieracitano et al., 2020): This paper presents an innovative intrusion detection model that integrates statistical analysis with an autoencoder-based deep learning approach, focusing on dimensionality reduction and overall performance improvement.

b. Trend Analysis

• Model Evolution: The progression moves from traditional ANN models [9] to hybrid models integrating autoencoders and SVM [2], [10].

• Focus on Dimensionality Reduction: Increasing use of autoencoders for feature extraction and data compression [2], [10].

• Dataset Consistency: NSL-KDD data set contains no duplicate data on both train and test data sets, unlike KDD CUP 99 [9].

I. Summary of review and research gap

After a detailed review of some of the related research papers, some important research gaps are identified, which are summarized below in the next section.

a. Summary of review

Performance analysis of NSL-KDD dataset using ANN evaluated the performance of an Artificial Neural Network (ANN) on the NSL-KDD dataset, highlighting the impact of feature selection on classification accuracy [9]. Deep learning approach combining sparse autoencoder with SVM for network intrusion detection proposed a hybrid deep learning model combining a Sparse Autoencoder for feature extraction with an SVM for classification, achieving significant accuracy improvements [2]. An empirical study on multiclass classification-based network intrusion detection [4]compared multiple computational intelligence techniques, including Decision Trees and Random Forest, for multiclass network intrusion detection, demonstrating their effectiveness on real-world datasets. A novel statistical analysis and autoencoder-driven intelligent intrusion detection approach developed an intrusion detection model that integrated statistical analysis with an autoencoder-based deep learning approach, effectively reducing dimensionality and enhancing performance [10].

b. Research gaps

Despite these advancements, several gaps remain. There was limited exploration of advanced deep learning models such as CNN, LSTM, or hybrid architectures. Feature engineering techniques were underexplored, and few studies performed comprehensive hyperparameter tuning or real-time detection evaluations. Additionally, issues like class imbalance and model explainability received minimal attention, and no studies addressed adversarial attack resilience, highlighting the need for more robust, scalable, and interpretable intrusion detection solutions.

3. Methodology

A. Deep Learning Techniques for Modeling Intrusion Detection Systems

In this proposed hybrid variational autoencoder-DenseNet convolutional neural network-based deep learning model, an unsupervised variational autoencoder neural network, along with a supervised DenseNet convolutional neural network, is going to be used to detect known as well as unknown complex malicious attacks on information systems. The proposal is designed the detect the possible threats to enterprises' ICT infrastructure network. Attackers can attack the enterprise's network and can possibly harm vital data of enterprises; making the network busy doing DoS attacks, can hamper the CIA triads of information of the enterprises. The application model of the proposed variational autoencoder-DenseNet convolutional neural network-based intrusion detection system in enterprise networks is shown in Figure 8.

In this proposed model, unsupervised variational autoencoder neural networks are trained with normal data, defining a threshold for reconstruction error, and a DenseNet convolutional neural network is trained with full normal as well as attack data. When both neural networks are trained, firstly, training data will be fed to the DenseNet convolutional neural network to classify input data. If DCNN classifies the input as an anomaly, then it will provide the result of the attack, and no further prediction is done using VAE. Secondly, if the prediction of DCNN is normal, then VAE prediction is done, where VAE calculates reconstruction error and compares it with the prescribed threshold. If the reconstruction error is greater than the threshold, then the model will predict it as an attack, and if the reconstruction error is less than or equal to the threshold, then the model will predict it as normal data. In this way, this hybrid model predicts the attack from the dataset.

B. Experimental Design and Setup

The proposed model for intrusion detection using NSL-KDD Train+ and Test+ data follows a structured pipeline comprising data loading, preprocessing, model training, and evaluation. First, the NSL-KDD Train+ and Test+ datasets are loaded, and only normal data from Train+ is separated to train the Variational Autoencoder (VAE). During preprocessing, categorical features are converted to numerical formats, and standard statistics like mean and median are computed to support reconstruction error calculations. The VAE model is then defined using an encoder-decoder architecture with ReLU and

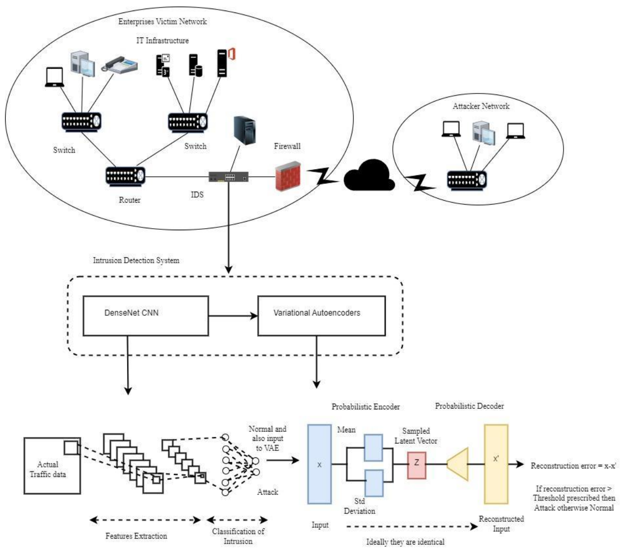

Figure 8. Application of Intrusion Detection System Model in Enterprise Networks.

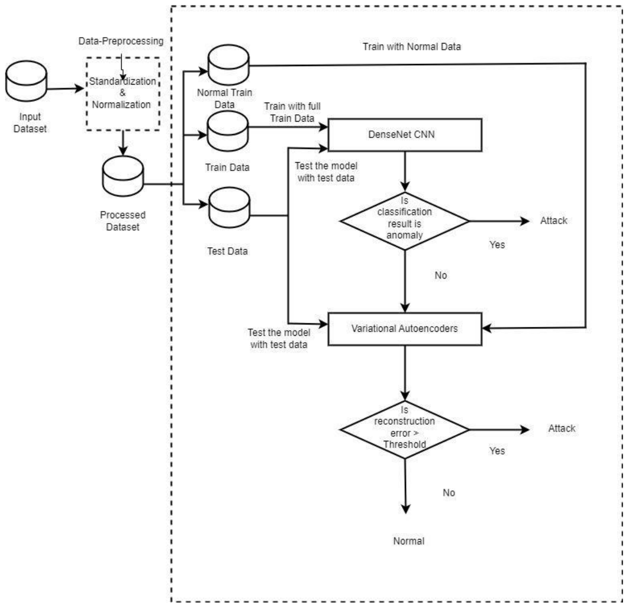
Data-Preprocessing
Standardization & Normalization
Input Dataset
Processed Dataset
Normal Train Data
Train Data
Test Data
Train with Normal Data
Train with full Train Data
DenseNet CNN
Test the model with test data
Is classification result is anomaly
Attack
Yes
No
Test the model with test data
Variational Autoencoders
Is reconstruction error > Threshold
Attack
Yes
No
Normal

Figure 9. Learning and Testing Process of the Hybrid Variational Autoencoder-DenseNet convolutional Neural Network Model

Sigmoid activation functions, and its latent vector is computed via a sampling layer. Hyperparameters are optimized using the Hyperband algorithm before training the VAE with normal data. Next, the VAE's reconstruction error is computed on the test set, with a 95th percentile threshold determining anomalies. The DenseNet CNN model, defined with batch normalization, ReLU, and Sigmoid activations, is also trained using the full processed training data, with its hyperparameters tuned via Hyperband. The hybrid model evaluates inputs sequentially: if DenseNet classifies data as anomalous, the result is finalized; otherwise, VAE verifies normal predictions using reconstruction error. Finally, the model's performance is assessed using metrics such as accuracy, precision, recall, F1-score, and false alarm rate (FAR) based on confusion matrix outcomes.

The basic overview of the workflow of this proposed model is as follows, and is described one by one as modeled in Figure 9.

a. Data loading and splitting normal data from NSL-KDD Train+ data

NSL-KDD Train+ and Test+ data are loaded first in CSV format separately. Both datasets contain normal and attack data, so to train the VAE of this model, only the normal data of the train+ data is separated and saved in a variable.

b. Data preprocessing

After loading the train and test, and separating the normal data from the train data. Train data, test data, and separated normal data are preprocessed, making them suitable for the proposed model. In this preprocessing process, categorical data are converted to a numerical format, suitable for training and testing the model and numerical data is needed to calculate the confusion matrix and performance metrics. Standard mean and median are also used to calculate reconstruction error.

c. Define VAE and train VAE with normal train data only

After preparing data suitable for the model, the VAE is defined and trained with normal training data. Before defining the VAE at first the latent vector is first calculated by a sampling layer, and then the VAE is defined using TensorFlow Keras. Encoder, decoder, and reconstructed data methods of the model are defined here using hyperparameters, and also define the encoder using relu activation function, defining the layers' density similarly. The decoder is defined using ReLU and sigmoid activation functions, defining the layers' density. After defining encoder, decoder, and latent vector, finally, the hyperband optimization algorithm is used for tuning the hyperparameter defined for VAE and searching for the best optimized parameters of VAE, and training the optimized VAE model with processed train normal data.

d. Determination of threshold and calculation of reconstruction error of VAE

The test of the VAE is done using processed data obtained after preprocessing of the test data, after defining and training the VAE model. During testing, the reconstruction error is obtained, and a threshold is specified in percentiles. In this paper, I use 95 percentile. The reconstruction error is compared with the threshold prescribed and determines whether it is an anomaly or not. If reconstruction error is greater than the threshold, then it is an anomaly; if it is less than the threshold, then it is not. Reconstructed data are obtained here to test the CNN model next.

e. Define DenseNet CNN and train DenseNet CNN with full training data

The DenseNet CNN model is defined using the hyperparameters of the model using Batch normalization, ReLU activation function for the pooling layer, and sigmoid activation function for the fully connected layer. After defining the model, the DenseNet CNN hyperparameter is tuned using a hyperband optimization algorithm with having epoch size of 20, and finding the best parameters of the DCNN model and training the searched model with processed train data.

f. Hybrid evaluation of the proposed model and test the model with Test data

Here hybrid evaluation of VAE-DCNN is done. First, training data will be fed to the DenseNet convolutional neural network to classify input data. If DCNN classifies the input as an anomaly, then it will provide the result of an attack, and no further prediction is done using VAE. Second, if the prediction of DCNN is normal, then the VAE prediction is done, where VAE calculates reconstruction error and compares it with the prescribed threshold. If the reconstruction error is greater than the threshold, then the model will predict it as an attack, and if it reconstruction error is less than or equal to the threshold, then the model will predict it as normal data.

g. Calculation of performance metrics and confusion matrix of the proposed model

Here, the real performance result of the proposed model is calculated considering evaluation metrics like true positive, true negative, false positive, and false negative, which are further used for the calculation of the confusion matrix and performance measures like accuracy, precision, recall, F1-score, and false alarm rate (FAR).

C. Programming Language and Libraries/Frameworks Used

To train and test this proposed model, also to visualize experiment results, I use the “Python” programming language and its various libraries, such as:

- Pandas
- NumPy
- Matplotlib
- Scikit-learn
- Tensorflow
- Keras

D. Hardware and Software Environment Specifications

For this work, the hardware and software environment specifications I used are listed below:

1. Hardware:

- Windows 10 Pro 64-bit (22H2, Build 19045.4651)
- Intel(R) Core(TM) i5-2410M CPU @ 2.30GHz2.30 GHz
- 8.00 GB RAM

Virtual Hardware:

- Kaggle RAM 30GB
- Kaggle Accelerator: GPU P100
- Kaggle Memory 73GB

1. Software:

- Kaggle notebook
- Microsoft Visual Studio Code (User) version 1.89.1
- Microsoft Office (Word and Excel)
- Draw.io

4. Results and Discussion

A. Performance Metrics for IDS

To calculate various measures for evaluation of the proposed model, first, different metrics of performance are calculated, which are defined below [12], [23]. True Positive (TP):- A true positive happens when the IDS successfully identifies a genuine attack. False Positive (FP):- A false positive happens when the IDS mistakenly classifies normal behavior as an attack. True Negative (TN):- A true negative occurs when the IDS accurately recognizes normal behavior as non-intrusive. False Negative (FN):- A false negative occurs when the IDS fails to detect a real attack.

Above mentioned four metrics are used to obtain a confusion matrix and five different performance Evaluation indicators of the proposed model. The evaluation indicator reflects the performance of the model. The Evaluation indicators and their corresponding formula are discussed below.

Accuracy: - If the model accurately classifies all data samples, it indicates high accuracy. Conversely, low accuracy suggests that the model fails to correctly classify certain data samples, which can be expressed as:

Accuracy = (TP + TN)/(TP + TN + FP + FN)

Precision: - Precision is a performance metric in deep learning (and machine learning) that assesses the accuracy of positive predictions made by a model. It calculates the ratio of true positive predictions (correctly identified positive instances) to the total number of instances predicted as positive, which includes both true positives and false positives. The formula of precision is given below.

Precision = TP/(TP + FP)

Recall: - In the context of a confusion matrix, recall (sometimes referred to as sensitivity or the true positive rate) represents the fraction of actual positive instances that the model correctly identifies. It is calculated as the ratio of True Positives (TP) to the total number of actual positive instances (TP + FN). Recall reflects the model's ability to successfully detect positive cases.

Recall =TP/(TP + FN)

F1-score: - The F1-score is a metric that integrates both precision and recall into one value. It is especially helpful when a balance between precision and recall is required, particularly in situations where one metric is more critical than the other. The formula is:

F1-score = (2 × Precision × Recall)/(Precision + Recall)

False Alarm Rate (FAR):- The False Alarm Rate (FAR) evaluates the percentage of negative instances that are wrongly identified as positive. It is also referred to as the False Positive Rate (FPR). The formula for FAR is given below.

FAR = FP/(FP + TN)

B. Experiment Results

In this proposed model, a hybrid VAE-DCNN deep learning model, the NSL-KDD dataset is used for the experiment. Train+ and Test+ data, part of NSL-KDD data, are loaded in CSV format. After loading Train+ and Test+ data, Train+ data is processed to get Train+ normal processed data to train the VAE model. During data preprocessing train and test data are changed from categorical format to numerical.

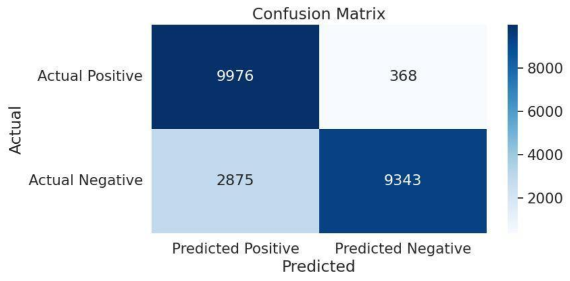

Figure 10. Confusion Matrix

Processed Train+, Test+ and Train normal data in numerical format are obtained after data preprocessing and used to train and test models and obtain the necessary evaluation metrics, like true positive, true negative, false positive and false negative to calculate the performance measure like accuracy, precision, recall, F1-score and false alarm rate (FAR) to check the efficiency and effectiveness of the model. After the experiment, the result for the evaluation metric is True positive = 9976, True negative = 9343, False positive = 368, False negative = 2857, which are essential to obtain performance parameters. Evaluation metrics are the basis for defining the model's effectiveness in detecting intrusion of network traffic. The confusion matrix is obtained using the above metric, which is shown in Figure 10.

Experiment results of this proposed model also give various performance measures results expressed in percentage, accuracy, precision, recall, F1-score, and false alarm rate (FAR), which are 85.7%, 96.44%, 77.74%, 86.09%, and 3.7% respectively. This model has an accuracy of 85.7% to detect anomalies correctly from the traffic and a 96.44% quality of positive prediction of the model. The experiment result is shown in Table 1 and also graphically represented in Figure 11.

Table 1. Experiment Result

Matric	VAE-DCNN Result (%)
Accuracy	85.7
Precision	96.44
Recall	77.74
F1 - Score	86.09
False Alarm Rate (FAR)	3.7

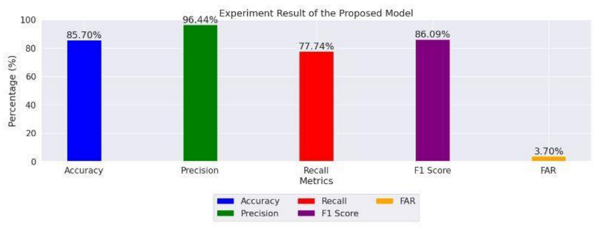

Figure 11. Graphical Representation of Experiment

C. Comparison with Other Related Works

Results obtained from the proposed model are compared with other related works done by other researchers using various deep learning models using NSL-KDD. I compared the results with deep learning models DNN, ANN, Sparse-AE+SVM, AE, LSTM, and MLP in the literature [2], [4], [9], [10], respectively.

As shown in the comparison table 2 given above, which is also visualized in a graphical presentation in Figure 12, the result of the proposed model is much better than other compared models.

Table 2. Comparison of Proposed Model Result with Other Related Works

Models	Accuracy	Precision	Recall	F1-Score
DNN [4]	85.74	96.73	77.19	86.05
ANN [9]	81.2	96.59	69.35	80.73
Sparse [2]	84.96	96.23	76.57	85.28
AE [10]	84.21	87.00	80.37	81.98
LSTM [10]	82.04	85.13	77.70	78.67
MLP [10]	81.65	85.03	77.13	78.67
VAE-DCNN	85.7	96.44	77.74	86.09

Accuracy: - The accuracy rate of 85.7% in this article is comparatively better than the accuracy rates in literature [2], [9], [10] using ANN, Sparse-AE+SVM, AE, LSTM, and MLP deep learning models, respectively, and equal to the accuracy rate in literature [4]. using a DNN deep learning model.

Precision: - The Precision rate of 96.44% in this proposed model is equal to the precision rate in the literature [2], [4], [9] using deep learning DNN, ANN, and Sparse-AE+SVM models, respectively, and better than the precision rate in the literature [10]using deep learning AE, LSTM, and MLP models.

Recall rate: - Recall rate 77.74% is the highest of all literature except literature [10] using a deep learning AE model.

F-score rate: - F-score rate 86.09 % is better than all literature [2], [4], [9], [10].

The bar graph given in Figure 12 shows a comparison of the performance metrics of various models, including DNN, ANN, Sparse-AE+SVM, AE, LSTM, MLP, and the proposed VAE-DCNN model, based on Accuracy, Precision, Recall, and F1-score.

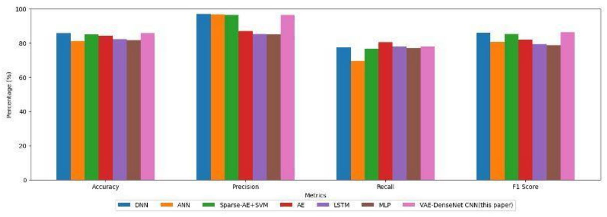

Figure 12. Graphical Representation of Comparison of Proposed Model Result with Other Related Works

Overall, the VAE-DCNN outperforms traditional models like ANN and MLP and hybrid models like Sparse-AE+SVM, showcasing its effectiveness in detecting intrusions with high precision and balanced recall.

D. Interpretation of Experimental Findings

The purpose of this research is to model an intrusion detection system for detecting known as well as unknown intrusions in the network, as well as in the host. The Experimental result of this proposed model is quite good. The accuracy rate of 85.7% of this model to predict anomalies is very good and makes the model good for intrusion detection, and the accuracy rate of this model is comparatively better than the accuracy rate in the literature [2], [4], [9], [10]. The precision rate of 96.44% of this model shows that the quality of prediction of true positives of this model is very high, which makes this model very reliable in the result. Recall rate of 77.74% of this paper indicates that among all the actual positives of the data, only 77.74% are classified as positives in the result of this paper, which is not satisfactory.F1 score rate of 86.09% of this model describes that the harmonic mean of precision and recall is quite good. Although the accuracy, precision, and F1 score rate of this proposed model are higher than 85% percent, which is a good result, it can be further improved to detect the intrusion correctly. The recall rate of this model is not satisfactory, so the recall rate should be increased to truly predict all positives among all possible positives of the dataset. The result of this experiment shows that known attacks are correctly predicted by this model, but 100% unknown attacks are not predicted while testing the model containing unknown attacks. Some percentage of unknown attacks is missed by this model to predict, so this problem needs to be corrected.

5. Conclusion

Cyber-attack and intrusions is a major problem in this big data era, causing a great threat to private as well as public data. Firewall, traditional method to detect threats is not sufficient to detect known, unknown, and sophisticated attacks, and is also not capable of filtering zero-day attacks, so to overcome this problem, various supervised, unsupervised, and hybrid deep learning techniques are used to model intrusion detection systems. In this paper hybrid VAE-DenseNet CNN deep learning model, VAE unsupervised technique, and DenseNet CNN supervised technique are used to filter attacks. To train and test the model NSL-KDD dataset is used to train and test the hybrid model. After the experiment satisfactory result was obtained with performance measure values accuracy = 85.7%, precision = 96.44%, recall = 77.74%, F1-score = 86.09%, and false alarm rate (FAR) = 3.7%. Comparing the results of this paper and other related work, DNN, ANN, Sparse-AE+SVM, AE, LSTM, and MLP show that performance, effectiveness, and efficiency are comparatively better than other related work.

The proposed model can be further improved by increasing the layers, increasing the number of convolutional layers, increasing the number of filters in it, increasing the number of max-pooling, increasing the filters of fully connected layers of CNN, by using different types of CNN, increasing and decreasing the density of encoders or decoders. Further different available datasets like UNSW–NB15, CICDDoS2019, etc. can be used to train and test the model.

A. Limitations of the Research

As we know that new intruders always use new kinds of attack techniques to attack the IT infrastructure, and zero-day attacks are common these days. This proposed model is conducted using only the NSL-KDD data set, which doesn't contain zero-day attack patterns and new kinds of attacks preferred by intruders, so this model can't predict zero-day attacks.

B. Future Research Directions

In this proposed model, VAE-DCNN, variational autoencoders, and DenseNet convolutional neural networks are used for hybrid prediction, in which a hyperband hyperparameter tuner is used for finding the best parameters of both models. ReLU activation function is used in AE, and ReLU is used in the convolutional layer, and sigmoid activation function in the fully connected layer is used in the DCNN model.

Further, other types of autoencoders and convolutional neural networks can be used for hybrid models with other types of hyperparameter tuners and activation functions, and another dataset like UNSW-NB15, CICDDoS2019 can be used to train and test the model.

Future Enhancement

Looking ahead, there are several exciting directions for enhancement. First, we plan to expand the library of supported components to include capacitors, inductors, and transistors. This would require the simulation engine to handle AC analysis and transient response, a significant but achievable challenge. Second, we intend to implement a save/load feature, allowing students to work on complex circuit designs over multiple sessions. Third, adding user accounts and a cloud-based system could enable instructors to assign circuits and track student progress. Finally, exploring more advanced visualization techniques, such as showing voltage levels through color gradients on the wires or using particle systems to more vividly represent electron flow, could further enrich the learning experience. In conclusion, the AR Circuit Simulator developed in this work is more than just a software application; it is a step towards a more dynamic, engaging, and effective paradigm for engineering education. By leveraging the power of augmented reality, we can transform the way students learn about and interact with the fundamental building blocks of modern technology.

Acknowledgment

I am pleased to acknowledge my thesis supervisor, Asst. Prof. Dr. Babu R. Dawadi, who provided me with the critical directions and valuable suggestions throughout my thesis study. Due to his support, I am successful in accomplishing my thesis work.

I would like to thank the Department of M.Sc. for providing me with the opportunity for this thesis work for the partial fulfillment of my master's degree in Information Systems Engineering. In addition, I would like to extend my sincere gratitude to the Asso. Prof. Parishwar Acharya, MSc. Program co-coordinator, for his kind cooperation and suggestions throughout my thesis work. Similarly, my special gratitude goes to Prof. Pradip Kumar Poudyal, Principal, AEC, and the whole college for providing me the opportunity to make some effort in the field of research.

Last but not least. I would like to express my heartfelt thanks to all my family and friends for their support and encouragement in accomplishing this research work.

Conflict of Interest

The authors declare no conflict of interest.

References

[1] A. R. Al-Ghuwairi and J. Al-Muhtadi, "Intrusion detection in cloud computing based on time series anomalies," *Journal of Cloud Computing*, 2023, doi: 10.1186/s13677-023-00491-x.

[2] M. Al-Qatif, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, "Deep learning approach combining sparse autoencoder with SVM for network intrusion detection," *IEEE Access*, vol. 6, pp. 52843–52856, 2018, doi: 10.1109/ACCESS.2018.2869577.

[3] Breach Level Index, "Data breach statistics," Nov. 2017. [Online]. Available: http://breachlevelindex.com/.

[4] W. Elmasry, A. Akbulut, and A. H. Zain, "Empirical study on multiclass classification-based network intrusion detection," *Computational Intelligence*, vol. 35, no. 4, pp. 915–954, 2019, doi: 10.1111/coin.12220.

[5] V. Hnamte and J. Hussain, "Network intrusion detection using deep convolution neural network," in *Proc. 2023 4th Int. Conf. Emerging Technology (INCET)*, 2023, pp. 1–6, doi: 10.1109/INCET57972.2023.10170202.

[6] R. Holdbrook and J. Smith, "Network-based intrusion detection for industrial and robotic systems," *Electronics*, vol. 13, no. 22, p. 4440, 2024, doi: 10.3390/electronics13224440.

[7] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 4700–4708, doi: 10.1109/CVPR.2017.243.

[8] "Information and communication technology (ICT) utilization and infrastructure alignment in construction organizations," ResearchGate. [Online]. Available: https://www.researchgate.net/.

[9] B. Ingre and A. Yadav, "Performance analysis of NSL-KDD dataset using ANN," in *2015 Int. Conf. Signal Process. Commun. Eng. Syst.*, 2015, pp. 92–96, doi: 10.1109/SPACES.2015.7058223.

[10] C. Ieracitano, A. Adeel, F. C. Morabito, and A. Hussion, "A novel statistical analysis and autoencoder driven intelligent intrusion detection approach," *Neurocomputing*, vol. 387, pp. 51–62, 2020, doi: 10.1016/j.neucom.2019.11.016.

[11] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 16–24, 2013, doi: 10.1016/j.jnca.2012.09.004.

[12] G. Liu and J. Zhang, "CNID: Research of network intrusion detection based on convolutional neural network," *Discrete Dynamics in Nature and Society*, Article 4705982, pp. 1–11, 2020, doi: 10.1155/2020/4705982.

[13] N. Mansouri and Z. Lachiri, "Laughter synthesis: A comparison between variational autoencoder and autoencoder," in *Proc. 2020 Int. Conf. Adv. Technol. Signal Image Process. (ATSIP)*, 2020, pp. 1–6, doi: 10.1109/ATSIP49331.2020.9231607.

[14] V. Z. Mohale and R. Kumar, "Evaluating machine learning-based intrusion detection systems with explainable AI," *Frontiers in Computer Science*, 2025, doi: 10.3389/fcomp.2025.1520741.

[15] A. Naghib and X. Zhang, "A comprehensive and systematic literature review on intrusion detection systems in the Internet of Medical Things," *J. Ambient Intell. Humanized Comput.*, 2025, doi: 10.1007/s10462-024-11101-w.

[16] K. Rajasekaran and K. Nirmala, "Classification and importance of intrusion detection system," *Int. J. Comput. Sci. Inf. Security*, vol. 10, no. 8, Aug. 2012.

[17] C. Rajathi, "Hybrid learning model for intrusion detection system," *J. King Saud Univ. - Comput. Inf. Sci.*, 2025, doi: 10.1016/j.jksuci.2024.12.001.

[18] N. Sainis, D. Srivastava, and R. Singh, "Classification of various dataset

for intrusion detection system," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 8, no. 1, pp. 50–56, 2018. [Online]. Available:

[19] M. Sajid and Y. Zhang, "Enhancing intrusion detection: A hybrid machine and deep learning approach," *Journal of Cloud Computing*, 2024, doi: 10.1186/s13677-024-00685-x.

[20] I. H. Sarker, "Deep cybersecurity: A comprehensive overview from neural network and deep learning perspective," *SN Computer Science*, vol. 2, no. 3, pp. 1–16, 2021, doi: 10.1007/s42979-021-00516-x.

[21] A. H. Sung and S. Mukkamala, "Identifying important features for intrusion detection using support vector machines and neural networks," in *Proc. Symp. Applications and the Internet*, 2003, pp. 209–216, doi: 10.1109/SAINT.2003.1183050.

[22] Symantec, "Internet security threat report 2017," vol. 22, Apr. 2017. [Online]. Available: https://www.symantec.com/content/dam/symantec/docs/reports/istr-22-2017-en.pdf.

[23] C. Zhang *et al.*, "A deep learning approach for network intrusion detection based on NSL-KDD dataset," in *Proc. 2019 IEEE 13th Int. Conf. Anti-counterfeiting, Security, and Identification (ASID)*, 2019, pp. 41–45, doi: 10.1109/ICASID.2019.8925239.

[24] Y. Zhang, "A review of deep learning applications in intrusion detection systems," *Electronics*, vol. 15, no. 3, p. 1552, 2025, doi: 10.3390/electronics15031552.