SSRG International Journal of Computer Science and Engineering
ISSN: 2348–8387 / https://doi.org/10.14445/23488387/IJCSE-V10I11P106

Volume 10 Issue 11, 43-48, November 2023

Original Article

A Deep Learning Approach for Intrusion Detection

T. Sai Harshitha[1], V. Sreenidhi[2], Sk. Parveen[3], P Tejaswini[4], Yerininti Venkata Narayana[5]

[1,2,3,4,5]*Department of Information Technology, Vasireddy Venkatadri Institute of Technology, Pedakakani (M), Guntur district, Andhra Pradesh, India.*

[1]*Corresponding Author : thotasaiharshitha20@gmail.com*

Received: 19 September 2023 Revised: 28 October 2023 Accepted: 10 November 2023 Published: 30 November 2023

Abstract *- In the ever-evolving field of cybersecurity, where protecting against unauthorized activities and emerging cyber threats is of paramount importance, this study explores the use of Convolutional Neural Networks (CNNs) to enhance intrusion detection capabilities. Traditional methods often struggle to keep pace with the increasing complexity and diversity of cyber threats, leaving organizations susceptible to data breaches, service disruptions, and financial losses. To address these challenges, CNNs, renowned for their feature extraction capabilities in image analysis, were adapted to the domain of network traffic analysis for precise intrusion detection. Also, conducted a comparative analysis, comparing the CNN approach against Autoencoders, a widely-used unsupervised learning technique for anomaly detection and evaluated the performance of CNN and autoencoders using metrics like accuracy, precision, recall, F1-Score, and AUC-ROC. The current study incorporates the Simargyl2022 dataset to enhance the quality of our results and analyses. This evaluation reveals the strengths and weaknesses of each technique, empowering cybersecurity professionals to make informed decisions about their intrusion detection systems, ultimately strengthening defenses against the ever-evolving cyber threat landscape and ensuring a safer digital world.*

Keywords *- Cybersecurity, Intrusion Detection System (IDS), Convolutional Neural Networks (CNN), Deep Learning, Malware, Port Scanning, Denial of Service (DoS), Network Security, Anomaly Detection, Cyberattacks.*

1. Introduction

In today's digital age, cybercrime has become a significant concern. Cybercrime refers to the various illegal activities that happen online. These activities can range from stealing sensitive information to disrupting online services and even committing financial fraud. An attacker's objective is to avoid detection by IDS by behaving like a regular user. Nevertheless, the patterns of their intrusive actions can differ, mainly because their intent may be specific, such as attempting unauthorized access to the machine and network.[1] In the digital age, cyberattacks are a major problem, and strong cybersecurity solutions are required to catch these attacks effectively. The good news is that advancements in machine learning, especially deep learning, now have better automated tools to detect and respond to these intrusions with greater accuracy. [2][3] Some common forms of cybercrime focus on DOS, malware, and port scanning. Denial of Service (DoS) attacks can disrupt websites and make them slow or completely unresponsive, causing frustration and potential financial losses for businesses. Denial-of-Service (DoS) attacks focus on services like emails and websites. They do this by overwhelming the network server with excessive traffic, causing it to become unresponsive.[3] Malware, short for "malicious software," is like a digital infection that can damage or steal information from computers with the unauthorized purpose of gaining access to sensitive data and compromising computer systems. Port scanning, conversely, involves sneaky attempts to find weaknesses in computer systems that hackers can exploit. There are some other intrusion attacks also; for example, in the paper [4], for detecting distributed denial of service attacks (DDoS) used an autoencoder (AE) based IDS

These cyber-threats keep changing and evolving, making it challenging for traditional security methods to keep up. That is where our project comes in. Network intrusion detection is crucial for protecting computer systems and networks from cyber threats. It monitors network traffic, identifying and preventing unauthorized access and malicious activities. By analyzing data patterns, intrusion detection systems swiftly detect suspicious behavior, providing early warnings to cybersecurity teams. This proactive approach is vital for preventing and mitigating cyber attacks ensuring the integrity and confidentiality of sensitive information in our interconnected digital landscape. Many intrusion detection algorithms based on deep learning have achieved good detection performance.[5] The current research uses Convolutional Neural Networks (CNN) to enhance our ability to detect and protect against these cyber threats. CNNs are powerful because they can automatically find important data patterns without manual help. CNNs are well-suited for image and sequence data analysis, making them an ideal choice for

processing network traffic data, which is inherently structured and can contain complex patterns. By adapting them to analyze network data, we aimed to improve our defenses against these ever-evolving online dangers, helping to keep our digital world safer. Many authors projected NIDS to extract feature representations followed by classifiers comprising AEs and/or deep belief networks.[6] Whereas Autoencoders are more versatile in handling various data types, they may not perform as well as CNNs on image or grid-like data. They lack the inherent ability to capture spatial hierarchies and local patterns that CNNs excel at in image-based anomaly detection. Several intrusion detection algorithms that utilize deep learning methods have shown impressive performance in detecting intrusions. [7]

2. Literature Review

In prior research, Zavrak et al. (2020) underscored the pivotal role of flow-based intrusion detection in effectively managing increased network traffic and uncovering unknown attacks through anomaly-based methodologies. Deep learning techniques, notably Autoencoder and Variational Autoencoder, have been employed to address key challenges, including dimensionality reduction and detecting of anomalous activities. Nevertheless, earlier investigations primarily revolved around content-based features and relied on outdated datasets, such as KDDCUP99. This current study stands out by strongly emphasizing flow-based data and harnessing the latent of deep learning for intrusion detection, with the VAE-based system outperforming alternative approaches.[7]

The paper by Yan et al. (2020) presents an innovative approach to IDS utilizing the techniques of deep learning, which include Autoencoders and Long Short Term Memory (LSTM) cells. This system is designed to address the evolving challenges of dynamic network traffic. The study underscores the crucial role of IDS in bolstering cybersecurity efforts and classifies various IDS methods. It strongly advocates for the application of deep learning, specifically Autoencoders and LSTM units, for feature extraction and classification, with the aim of achieving superior accuracy.[8]

Previous research by W. Wang et al. has addressed security concerns in cloud computing by developing intrusion detection systems. This involved SDN routing traffic to intrusion detection systems and using shallow machine learning techniques. These techniques faced feature dimensionality and performance issues, prompting the exploration of deep learning, particularly autoencoders, for effective feature extraction in intrusion detection systems.[9]

The study by B. Min. highlights the restrictions in signature-based IDS and the growing reliance on anomaly detection through machine learning, especially One-Class Learning methods like Support Vector Machine and AE. The over-generalization problem in Autoencoder-based models is a recognized issue that the proposed MemAE model addresses. This survey underscores the need for innovative approaches to combat evolving network intrusion threats.[10]

The paper by W. Hui et al. [11] presents a novel IDS for Power IoT, emphasizing anomaly detection. It combines autoencoder networks and GMM to address the shortcomings of unsupervised models. The proposed architecture enhances accuracy and F1-score in intrusion detection while efficiently handling data collection, processing, and feature extraction.

3. Methodology

3.1. Data Collection and Data Loading

- In this first step, the network traffic data is collected from various sources, such as log files or network capture tools. This data serves as the foundation for building an intrusion detection system.
- One common approach to loading this collected data is using the Pandas library in Python, which offers potent tools for data exploitation, making it a preferred choice for handling structured data like network traffic logs. Fig 1 below shows the Methodology of intrusion detection using CNN.

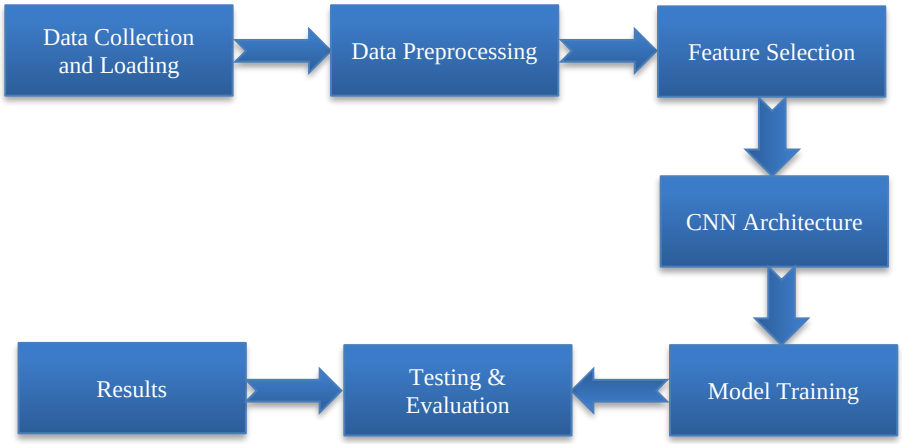

Fig. 1 Methodology of intrusion detection using CNN

3.2. Data Preprocessing

After loading the data, the next step is data preprocessing, which is essential to prepare the data for analysis and model development. Data preprocessing includes tasks like handling missing values, removing outliers or noisy data points, and standardizing features. These operations ensure that the data is in a clean and usable format.

For instance, remove columns with missing data using data.dropna()and handle infinite values with data.replace([np.inf, -np.inf], np.nan). This ensures that the data is free from irregularities that might affect the model's performance. When dealing with categorical variables, the process of one-hot encoding is employed to transform them into integer representations.[12]

3.3. Feature Selection

3.3.1. Feature Selection

In the context of intrusion detection, not all attributes in the network traffic data are equally informative. Some attributes may contain redundant or irrelevant information. Feature selection is employed to identify and retain the most relevant attributes, improving the model's accuracy and efficiency.

3.3.2. SelectKBest and Feature Scoring

The SelectKBest method, available within the scikit-learn library, is utilized for feature selection. It relies on statistical techniques, such as the F-statistic, to score each feature's importance. Features with higher scores are considered more valuable for the intrusion detection task.

Among all the features, the best features are selected and are shown below in Fig 2.

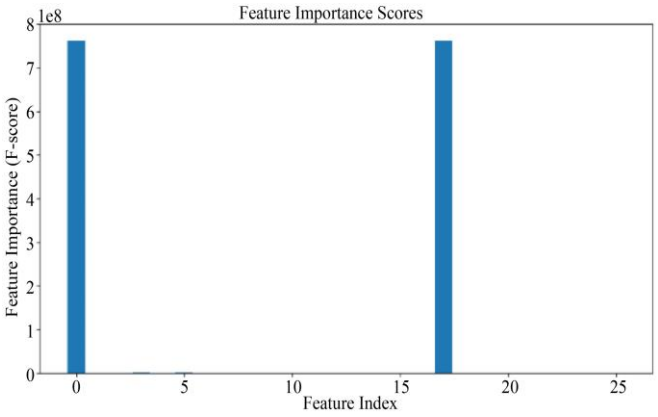

Fig. 2 Best features

3.3.3. Data Reshaping

Following feature selection, the data is reshaped to include only the chosen top features. This reshaping ensures that the subsequent intrusion detection model works with a streamlined set of attributes, enhancing its efficiency and effectiveness.

3.4. CNN Model Architecture

3.4.1. Input Layer

It receives the raw data, which is often an image or a volume of data. Each neuron in the input layer corresponds to a pixel or a feature in the input data.

3.4.2. Convolutional Layers

Convolutional layers are at the heart of CNNs. These layers apply convolution operations to the input data. Convolution involves sliding small filters (also called kernels) over the input data to detect features like edges, corners, and textures. Convolutional layers have multiple filters to extract different features simultaneously. Each filter's output is referred to as a feature map. After each convolution operation, an activation function (ReLU), commonly used as the activation function, is applied. It introduces non-linearity by converting all negative values to zero and keeping positive values as they are.

3.4.3. Pooling Layers

Pooling layers serve to diminish the spatial dimensions of feature maps. A frequently applied method is max-pooling, in which a small region (e.g., 2x2) is examined, and only the maximum value is retained, while the others are discarded. Pooling helps reduce the computational load and provides translational invariance.

3.4.4. Fully Connected Layers

Here, they connect all neurons in one layer to all neurons in the previous layer. These layers are used for high-level feature extraction and classification. Typically, there are one or more fully connected layers in the architecture.

3.4.5. Flatten Layer

Before entering the fully connected layers, the output from the previous layers (usually 2D feature maps) is flattened into a 1D vector. This transformation allows the feature maps to be processed by the fully connected layers.

3.4.6. Output Layer

This layer is responsible for generating the ultimate predictions or classifications. The quantity of neurons in this layer aligns with the total number of classes or categories relevant to the given problem. Also, it provides the final classification. Neurons in this layer produce class probabilities. In the case of intrusion detection, you may have one neuron per class (e.g., one for "normal" and others for different intrusion types). The neuron with the highest probability indicates the predicted class. Activation functions like softmax are used to obtain class probabilities. Below fig. 3 is the example diagram for CNN Architecture.

3.5. Model Training

In this phase, the data is divided into two parts, namely, the training set and the testing set. Employ K-Fold Cross-validation for the purpose of training and validation. Training

the CNN model on the training dataset. The model learns to differentiate between normal and malicious network traffic during training.

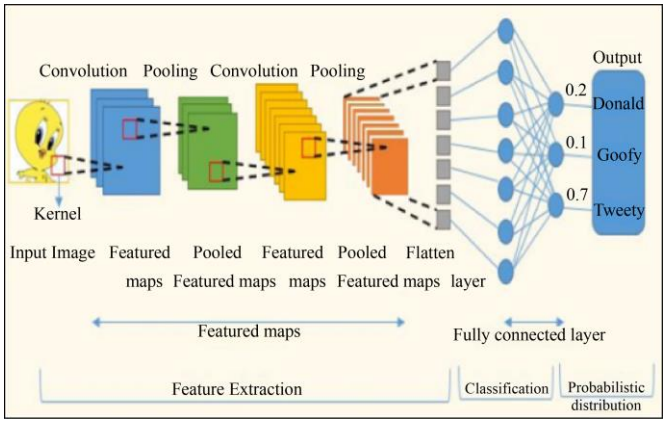

Fig. 3 Example for CNN Architecture[13]

3.6. Testing and Evaluation

Plot the accuracy of the model during training. Load the best model weights. Evaluate the model on the test set to obtain test accuracy. Testing the trained CNN on the testing dataset. When evaluating the models, provided standard metrics, including accuracy, true positive rate, and false positive rate, to enable a meaningful comparison between various model configurations. [14] Calculating performance metrics like accuracy, precision, recall, and F1-score. Figure 4 below shows the architecture of intrusion detection using CNN. The proposed architecture indicates the strategies suggested for our experiment.[15] Therefore, it is affirmed that the accuracy of the current study performance surpasses prior studies in Network Intrusion Detection Systems (NIDS), which employed unsupervised learning techniques, specifically clustering algorithms.[16]

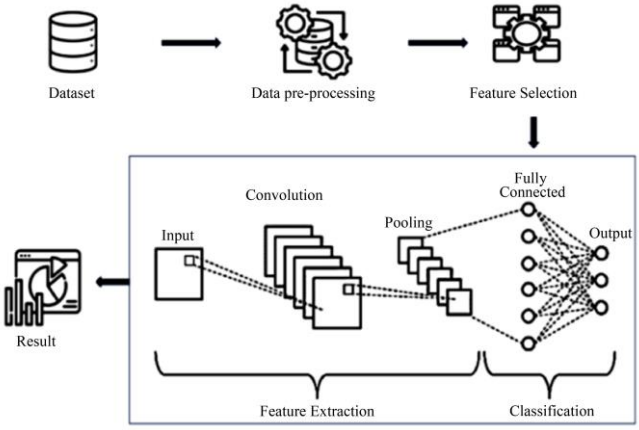

Fig. 4 Architecture of intrusion detection using CNN

4. Results

4.1. Performance Analysis

Table 1 compares the performance metrics for the IDS on the Simargyl dataset using Convolutional Neural Networks (CNN) and Autoencoders. These metrics, including precision, recall, and F1-score, are crucial for assessing the effectiveness of our models. The results clearly indicate that the CNN outperforms the autoencoder across all metrics.

Table 1. Comparitive analysis for performance metrics

Performance Metrics	Algorithms	
	CNN	Autoencoders
Precision	96.12	93.33
Recall	96.12	100
F1_score	96.12	96.5

4.2. Visual Representation of Results

Visual Representation using barchart for the above metrics analysis is shown in Fig 5 below.

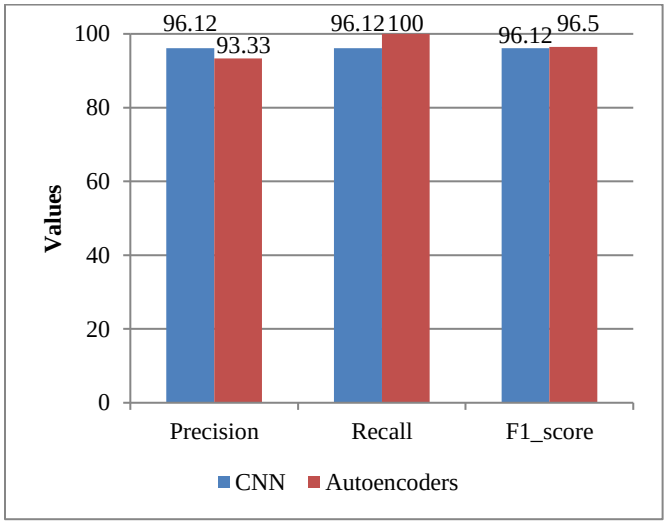

Fig. 5 Visual representation of performance metrics

These results and visual representations underscore the robustness and reliability of our CNN-based intrusion detection system over autoencoders, emphasizing its value in enhancing network security and safeguarding against various cyber threats.

The high accuracy and balanced Precision, Recall, and F-Measure metrics showcase the system's ability to distinguish normal from intrusive network traffic patterns effectively.

4.2.1. Confusion Matrix

In our evaluation of intrusion detection using Convolutional Neural Networks (CNN) on the Simargyl2022 dataset, a confusion matrix was constructed to provide a detailed view of the model's performance.

The confusion matrix allows us to understand how well the system differentiates between normal and intrusive network traffic by breaking down the results into four categories: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).

4.2.2. True Positives (TP)

These are instances where the system correctly identified intrusion attempts. In the context of Simargyl2022, these represent network traffic patterns correctly classified as intrusive.

4.2.3. True Negatives (TN)

TN instances are where the system correctly recognized normal network traffic. These are the patterns that are accurately classified as non-intrusive.

4.2.4. False Positives (FP)

FP occurs when the system mistakenly identifies normal traffic as intrusive. These are instances where the system raises an alarm without a real intrusion.

4.2.5. False Negatives (FN)

FN instances indicate that the system failed to identify intrusion attempts, leading to a false sense of security.

By analyzing the values in the confusion matrix, gained an insight into the system's effectiveness in distinguishing between different network traffic patterns, highlighting its capability to accurately identify intrusion attempts while minimizing the risk of overlooking potential threats. Figures 6 and 7 below are the confusion matrices for Autoencoders and CNN on simargyl2022.

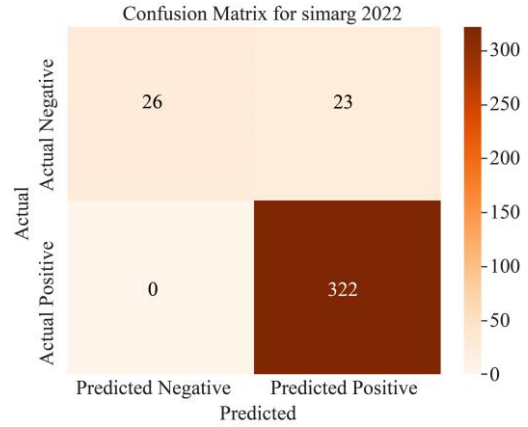

Fig. 6 Confusion matrix for autoencoders

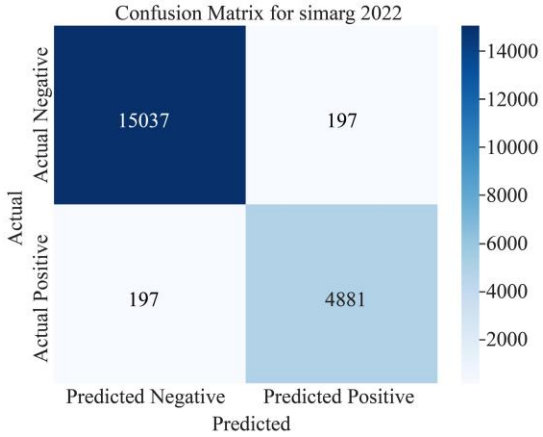

Fig. 7 Confusion matrix for CNN

4.3. Comparative Analysis of Intrusion Detection using CNN and Autoencoder

Table 2 compares the accuracy of the Intrusion Detection System (IDS) models, specifically Autoencoders and Convolutional Neural Networks (CNN), when evaluated on the Simargyl dataset. The accuracy metric serves as a fundamental measure of the model's overall performance. As depicted in the table, the CNN model achieves a notably higher accuracy of 98.06, whereas the Autoencoders attain an accuracy of 93.80. These results demonstrate a clear advantage for the CNN model regarding overall accuracy, reinforcing its suitability for the IDS task on the Simargyl2022 dataset.

Table 2.Comparative analysis of accuracy values

Performance	CNN	Autoencoders
Accuracy	98.06	93.80

4.4. Visual Representation of Accuracy Comparision

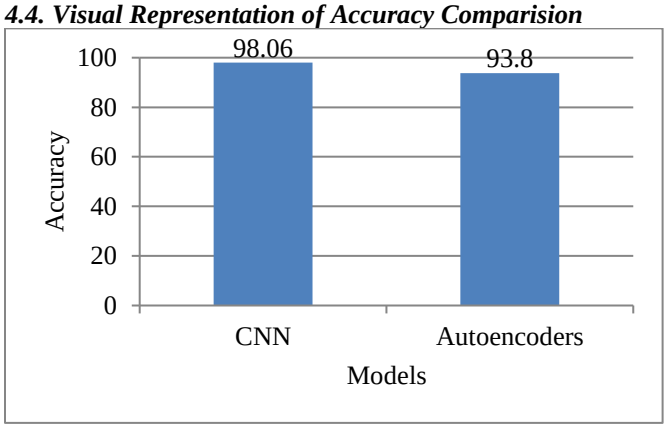

Fig. 7 Comparative analysis of accuracy values

This content provides a concise but informative table description, conveying the key findings to the reader. The pie chart can be referenced within the text or displayed immediately after the table to represent the same information in a graphical form visually.

5. Conclusion

In conclusion, Convolutional Neural Networks (CNNs) have shown their superiority in intrusion detection compared to traditional methods like autoencoders, resulting in higher accuracy and improved effectiveness. Implementing preventive measures is essential to strengthen network security further. This involves real-time response mechanisms, behavioral anomaly detection, adaptive firewalls, threat intelligence, User and Entity Behavior Analytics (UEBA), security patch management, employee training, and the development of detailed intrusion response plans. Continuous evaluation and adjustment are necessary to keep up with evolving threats, thereby enhancing the network's defenses against intrusion attempts and protecting digital assets. This transforms the security strategy into a vigilant protector that actively prevents and counters evolving cyber threats.

References

[1] Ghulam Muhammad, M. Shamim Hossain, and Sahil Garg, "Stacked Autoencoder-Based Intrusion Detection System to Combat Financial Fraudulent," *IEEE Internet of Things Journal*, vol. 10, no. 3, pp. 2071-2078, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[2] Yerininti Venkata Narayana, and Mooramreddy Sreedevi, "Deep Neural System for Identifying Cybercrime Activities in Networks," *Journal of Theoretical and Applied Information Technology*, vol. 101, no. 16, pp. 6414-6424, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[3] Fadi Aloul et al., "Network Intrusion Detection on the IoT Edge Using Adversarial Autoencoders," *2021 International Conference on Information Technology (ICIT)*, *IEEE*, pp. 120-125, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[4] Firuz Kamalov et al., "Autoencoder-based Intrusion Detection System," *2021 International Conference on Engineering and Emerging Technologies (ICEET)*, *IEEE*, pp. 1-5, 2021. [CrossRef] [Publisher Link]

[5] Ruijie Zhao et al., "An Efficient Intrusion Detection Method Based on Dynamic Autoencoder," *IEEE Wireless Communications Letters*, *IEEE*, vol. 10, no. 8, pp. 1707-1711, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[6] Kohei Shiomoto, "Network Intrusion Detection System Based on an Adversarial Auto-Encoder with Few Labeled Training Samples," *Journal of Network and Systems Management*, vol. 31, no. 5, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[7] Sultan Zavrak, and Murat İskefiyeli, "Anomaly-Based Intrusion Detection from Network Flow Features Using Variational Autoencoder," *IEEE Access*, vol. 8, pp. 108346-108358, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[8] Yu Yan et al., "A Network Intrusion Detection Method Based on Stacked Autoencoder and LSTM," *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, *IEEE*, pp. 1-6, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[9] Wenjuan Wang et al., "Cloud Intrusion Detection Method Based on Stacked Contractive Auto-Encoder and Support Vector Machine," *IEEE Transactions on Cloud Computing*, vol. 10, no. 3, pp. 1634-1646, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[10] Byeongjun Min et al., "Network Anomaly Detection Using Memory-Augmented Deep Autoencoder," *IEEE Access*, vol. 9, pp. 104695-104706, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[11] Wang Hui et al., "A Framework for Network Intrusion Detection Based on Unsupervised Learning," *2021 IEEE International Conference on Artificial Intelligence and Industrial Design (AIID)*, *IEEE*, pp. 188-193, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[12] Prabhav Gupta, Yash Ghatole, and Nihal Reddy, "Stacked Autoencoder based Intrusion Detection System using One-Class Classification," *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, IEEE, pp. 643-648, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[13] Ali Hakem Alsaeedi et al., "Dynamic Clustering Strategies Boosting Deep Learning in Olive Leaf Disease Diagnosis," *Sustainability*, vol. 15, no. 18, pp. 1-20, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] Youngrok Song, Sangwon Hyun, and Yun-Gyung Cheong, "Analysis of Autoencoders for Network Intrusion Detection," Sensors, vol. 21, no. 13, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[15] Srikanthyadav Moraboena, Gayatri Ketepalli, and Padmaja Ragam, "A Deep Learning Approach to Network Intrusion Detection Using Deep Autoencoder," *Revue d'Intelligence Artificielle*, vol. 34, no. 4, pp. 457-463, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[16] Hyunseung Choi et al., "Unsupervised Learning Approach for Network Intrusion Detection System Using Autoencoders," *The Journal of Supercomputing*, vol. 75, pp. 5597–5621, 2019. [CrossRef] [Google Scholar] [Publisher Link]