Advanced Network Threat Detection Using Deep Learning Models

# Khushi Kharate#1, Panchami Vishwakarma#2, Rama Labhe#3, Aryaman Pandey#4

Shri Ramdeobaba College of Engineering and Management Pincode 440013, Nagpur, India

***Abstract* — Object detection and recognition are important tasks in computer vision and are widely used in many fields. This article provides an overview and comparison of the main technologies used in the mission to detect, recognize and follow the changes transferred from modern methods to modern deep learning, starting from the historical background. This study compares and contrasts popular techniques such as field detectors, single-shot devices, and two-phase detectors, highlighting their advantages and disadvantages. In addition, recent developments are discussed, including one-off studies of monitoring systems and reform strategies, combining existing research and analysis control examples. This article is a useful resource for researchers to research and search for information.**

***Keywords— Network Intrusion ; Treat Detection ;* Deep Learning ; MLP ; LSTM ; Autoencoder ; Privacy and Security**

1. INTRODUCTION

With the rapid advancement of technology and the increasing dependence on digital infrastructure, cybersecurity has become a critical concern. One of the primary threats faced by organizations is network intrusion, where unauthorized users attempt to exploit vulnerabilities in a network system. Network Intrusion Detection Systems (NIDS) play a pivotal role in safeguarding against such attacks by monitoring network traffic and identifying suspicious activities in real-time.

This research paper focuses on the development of a robust network intrusion detection system using machine learning techniques. Traditional NIDS often rely on signature-based methods, which are effective but limited in detecting new or evolving attack patterns. To address this, our model incorporates machine learning algorithms that can learn from network data, recognize anomalous behavior, and detect potential intrusions more effectively.

The proposed system uses a combination of feature extraction techniques and supervised learning algorithms to classify network traffic as either normal or malicious. By evaluating various algorithms, including [mention any specific algorithms used in your project], we demonstrate the system's ability to achieve high accuracy in intrusion detection. Furthermore, the system is designed to be scalable and adaptable, ensuring it can evolve with changing network threats.

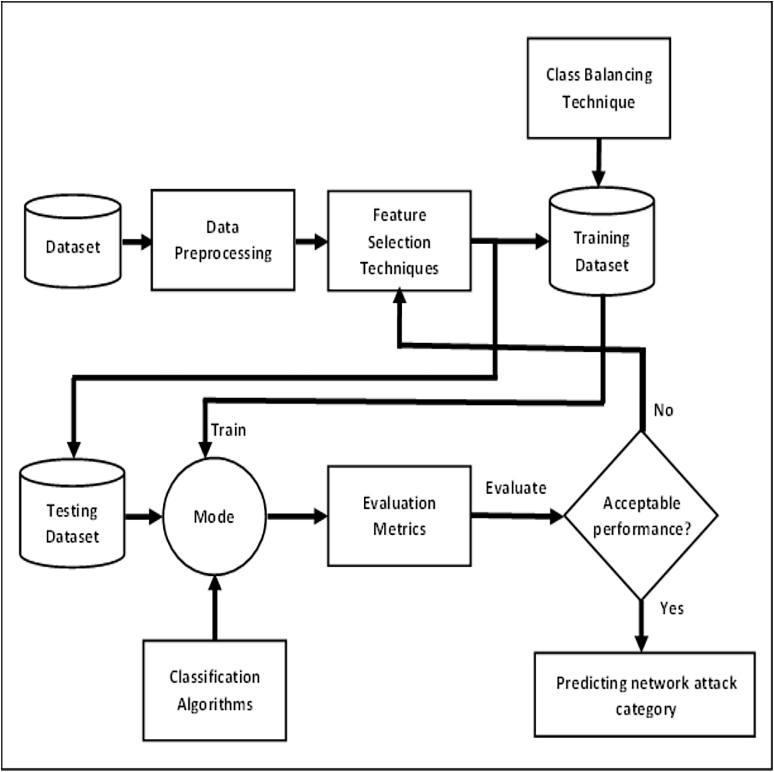


Fig. 1 Basic Network Intrusion Detection System

As cyberattacks become more sophisticated, traditional signature-based intrusion detection systems often struggle to keep up with emerging threats. Machine learning offers a solution by automatically learning from network data to detect both known and unknown attacks. In this research, we integrate machine learning algorithms into a network security framework to enhance detection accuracy and speed. This approach reduces reliance on manual updates and improves the system's ability to identify malicious activities in real-time, making networks more resilient against evolving threats.

This paper outlines the system's design, implementation, and performance evaluation, providing insights into how machine learning can enhance the detection capabilities of network intrusion systems.

1. HISTORICAL BACKGROUND

The concept of network intrusion detection began in the late 1980s with James P. Anderson's report and Dorothy Denning's development of the "Intrusion Detection Expert System" (IDES), which introduced detecting anomalous behavior in network activities. Early systems were signature-based, effective against known attacks but limited in identifying new threats. As cyberattacks became more sophisticated, researchers shifted toward anomaly detection and machine learning techniques in the 1990s and 2000s. This evolution from host-based to network-based systems enabled real-time analysis of network traffic, making intrusion detection systems more adaptive and effective in identifying unknown threats.

III . LITERATURE SURVEY

With Network intrusion detection is essential for safeguarding digital infrastructures, and deep learning methods have greatly advanced this field. Prior studies have employed various architectures to improve NIDS performance, such as Self-Taught Learning (STL) for flexibility, deep neural networks for high accuracy in classification, and recurrent architectures like LSTM and GRU to capture sequential attack patterns. While these approaches offer benefits—like adaptability for zero-day attacks or unsupervised learning with autoencoders—their focus often remains narrow, addressing either static classification or temporal patterns without a comparative evaluation across multiple models. In contrast, our research comprehensively compares three models—MLP, LSTM, and Autoencoder—on both binary and multi-class classifications using the NSL-KDD dataset. This approach captures the unique strengths of each model: MLP’s efficiency in standard classification, LSTM’s capability to learn sequential dependencies, and Autoencoder’s effectiveness in anomaly detection. By assessing each model’s performance on multiple metrics, including accuracy, precision, recall, and ROC-AUC, this study provides a holistic framework, addressing varying intrusion types and detection criteria more flexibly and effectively than existing works. This multifaceted comparison supports the development of a robust, adaptable NIDS framework capable of responding to a broader spectrum of attack scenarios.

1. **METHODOLOGY**

The development of the proposed network intrusion detection system (NIDS) utilizes deep learning techniques to improve detection accuracy. The methodology involves several key steps:

1. **Data Collection**

The dataset for this research is sourced from publicly available network traffic datasets, such as NSL-KDD or CICIDS2017, containing both benign and malicious traffic instances. These datasets offer diverse attack types like Denial of Service (DoS), Probe, and User to Root (U2R) attacks, providing a comprehensive foundation for model training and evaluation.

## **Data Preprocessing**

The raw dataset is preprocessed to remove noise and handle missing or inconsistent data. Categorical features are encoded, and numerical features are normalized to ensure they are on a comparable scale. To improve performance, outliers are removed, and data is split into training and testing sets to allow for model validation.

1. **Feature Extraction**

Feature extraction is performed to reduce dimensionality and enhance the learning process. Techniques such as Principal Component Analysis (PCA) or Recursive Feature Elimination (RFE) may be used to select the most relevant features for distinguishing between normal and malicious traffic. This step helps in reducing the computational complexity of the deep learning models.

**D. Label Encoding**

The NSL-KDD dataset includes labeled data with multiple classes representing different types of network activities, both normal and various forms of attacks. We converted the categorical attack labels into numerical formats for model compatibility. For **binary classification**, we grouped all attack classes under a single label ("abnormal") while retaining "normal" traffic as a separate label, thus distinguishing between benign and malicious activities. For **multi-class classification**, we retained the distinct attack categories, labeling each as a specific integer. This enabled the models to learn both general intrusion detection (binary) and nuanced attack identification (multi-class), improving the overall adaptability of the NIDS.

**E. Train – Test Spilt**

To ensure robust performance evaluation and avoid overfitting, the dataset was divided into training and testing sets in a 75%-25% ratio. The training set, comprising 75% of the data, was used for model training, enabling the models to learn the patterns and characteristics of both normal and attack data. The testing set, comprising the remaining 25% of the data, was used exclusively for evaluating model performance after training, ensuring an unbiased measure of each model's detection capability on unseen data.

Each model was trained independently on the preprocessed NSL-KDD dataset, following identical preprocessing steps to maintain comparability. This setup allowed us to benchmark each model—MLP, LSTM, and Autoencoder—on equal terms and analyze their respective strengths and limitations in handling network intrusion detection tasks across both binary and multi-class settings. The standardized preprocessing and model training pipeline provide a fair basis for evaluating and comparing model performance, specifically focusing on accuracy, precision, recall, F1-score, and ROC-AUC metrics.

**VI. MODEL SELECTION**

**1. Multi-Layer Perceptron (MLP)**  
The Multi-Layer Perceptron (MLP) serves as a foundational model for classification tasks within this research. The architecture consists of an input layer, a single hidden layer containing 50 neurons, and an output layer activated by the sigmoid function. The hidden layer employs the Rectified Linear Unit (ReLU) activation function, which introduces non-linearity into the model, allowing it to learn complex patterns in the data. MLPs are particularly effective in capturing non-linear relationships, making them well-suited for tasks where traditional linear models may fail. The capability to learn from a diverse set of features enables the MLP to classify network traffic effectively, distinguishing between normal behavior and potential attacks.

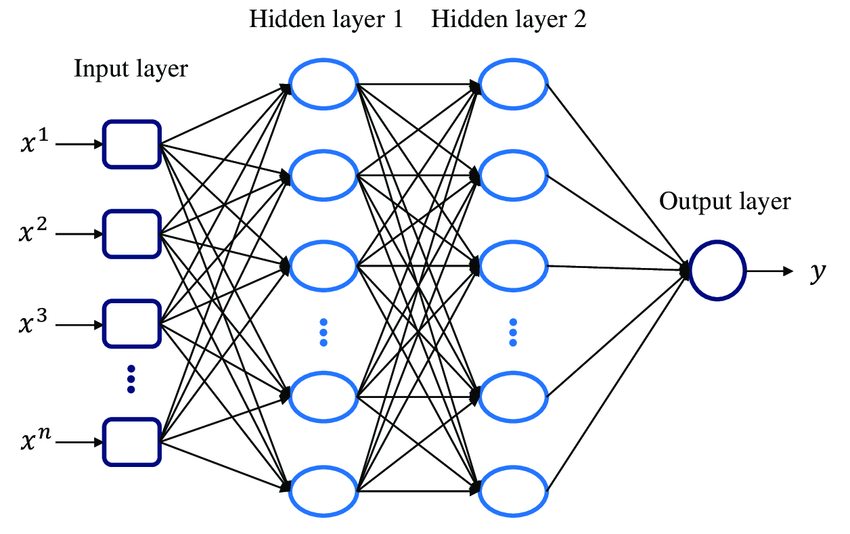


Fig 2 – MLP Architecture

**2. Long Short-Term Memory (LSTM)**   
The Long Short-Term Memory (LSTM) model was specifically designed to handle sequential data, making it ideal for capturing temporal patterns within the network data. The architecture includes a single LSTM layer comprising 50 neurons, followed by a sigmoid-activated output layer. LSTMs are equipped with memory cells that facilitate the retention of information over extended periods, allowing the model to learn dependencies across time steps. This characteristic is critical for identifying patterns in attack sequences, where the timing and order of events can provide valuable insights into potential threats. By leveraging the LSTM's ability to model temporal dependencies, this study aims to enhance the detection of complex attack patterns that may evolve over time.

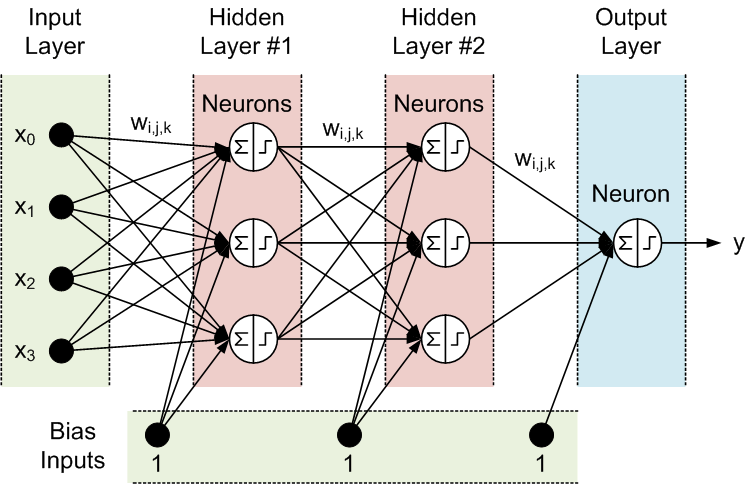


Fig 3 – LSTM Architecture

3.**Autoencoder**  
The Autoencoder is employed as an unsupervised learning approach to anomaly detection. Its architecture consists of an encoding layer with 50 neurons, followed by a decoding layer. The primary objective of the Autoencoder is to minimize reconstruction error—essentially learning to compress the input data into a lower-dimensional representation and then reconstruct it back to its original form. This model is particularly effective in identifying deviations from normal behavior, as it does not rely on predefined labels for training. By focusing on the reconstruction process, the Autoencoder can detect anomalies that manifest as significant differences between the input and its reconstruction, making it suitable for uncovering previously unseen threats within network data.

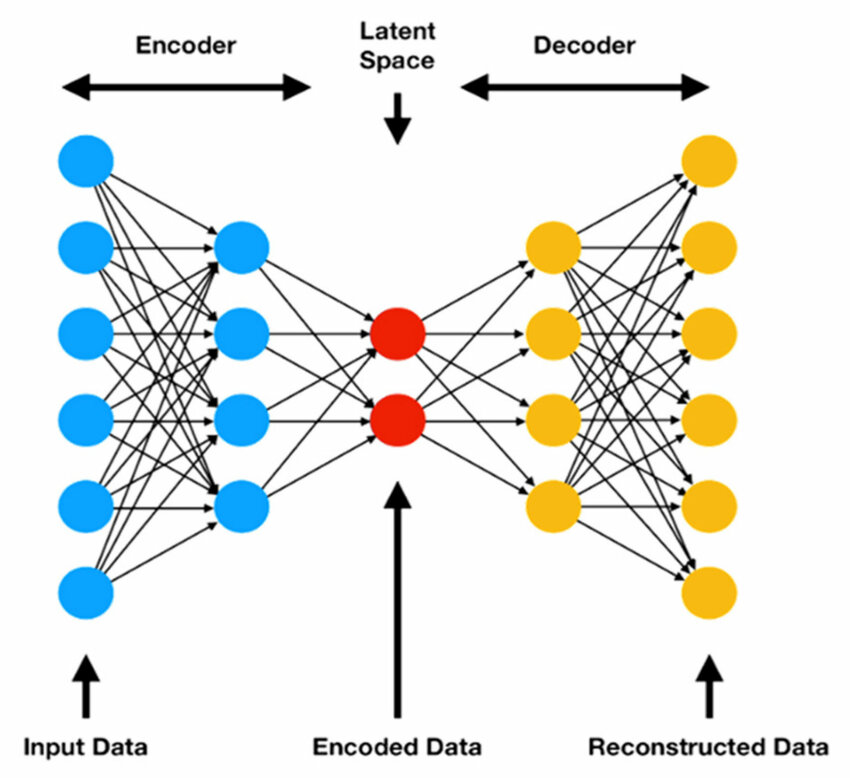


Fig 4 – Autoencoder Architecture

**V. RESULTS**

The performance of each deep learning model was evaluated using key metrics: accuracy, precision, recall, F1-score, and loss. These metrics provide a comprehensive view of each model’s effectiveness in detecting anomalies and attacks within network data. The summarized results are as follows:

Performance Metric

1. **Recall Score**: Measures the model’s ability to correctly identify all relevant instances, reflecting how well it minimizes false negatives.

Recall = TP / TP + FN

1. **F1 Score:** Represents the harmonic mean of precision and recall, providing a balanced measure of the model’s accuracy for imbalanced datasets.

F1-Score = 2\* Precision \* Recall / Precision + Recall

1. **Precision Score:** Indicates the model’s accuracy in predicting positive instances, showing its ability to minimize false positives.

Precision = TP / TP + FN

1. **MLP (MultiLayer perceptron)**

The MLP model achieved an accuracy of approximately 97% on the test set, with high precision, recall (0.986), and F1-score (0.979). Its fast convergence allowed it to effectively learn complex relationships, making it highly reliable for real-time classification tasks in network security.

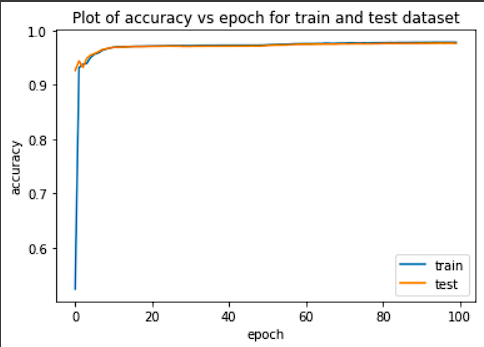


Fig5 - Plot of Accuracy vs epoch for train and test dataset

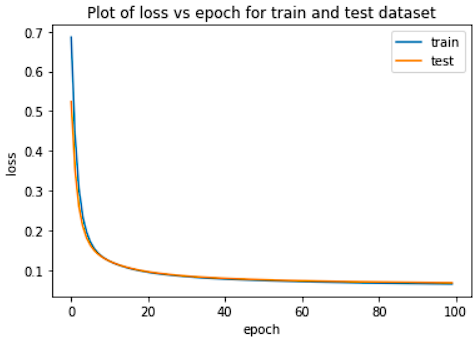


Fig 6 - Plot of loss vs epoch for train and test dataset

**2. LSTM (Long Short Term Memory)**

The LSTM model achieved an accuracy of around 97%, effectively capturing temporal patterns for sequential attack detection. Its high precision, recall (0.986), and F1-score (0.979) underscore its strength in minimizing false negatives, making it suitable for scenarios where event order is critical.

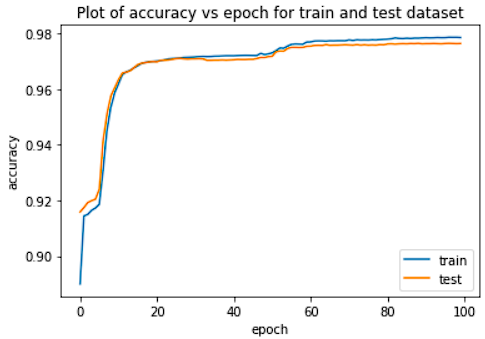


Fig.7 - Plot of Accuracy vs epoch for train and test dataset

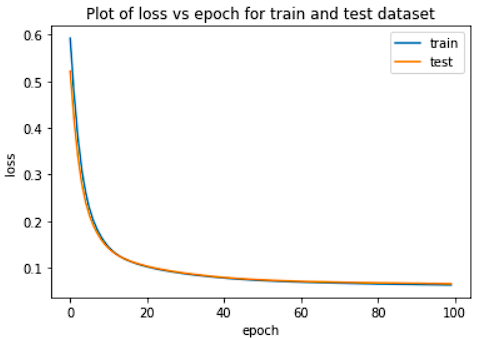


Fig.8 - Plot of loss vs epoch for train and test dataset

**3. Autoencoder**

The Autoencoder model achieved around 85% accuracy in anomaly detection, effectively identifying deviations in unsupervised settings. While it has limitations in labeled classification tasks, its recall (0.978) and F1-score (0.857) demonstrate its capability for detecting novel attacks.

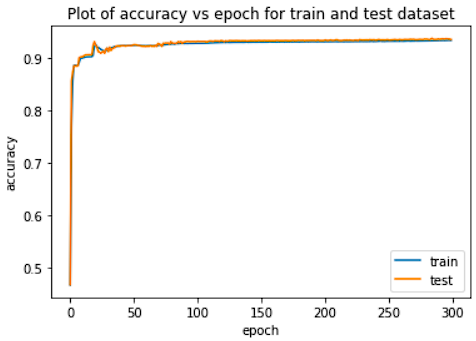


Fig 9 - Plot of Accuracy vs epoch for train and test dataset

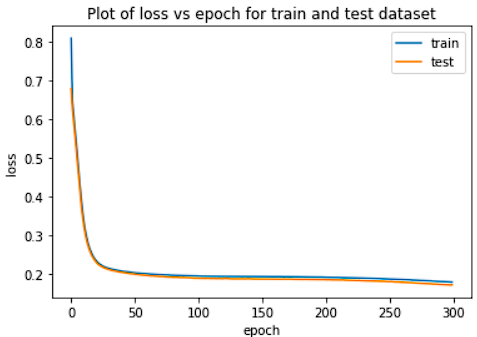


Fig. 10 - Plot of Loss vs epoch for train and test dataset

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | MLP | LSTM | Autoencoder |
| Accuracy | 97% | 97.7% | 82.6% |
| Recall Score | 0.986 | 0.98 | 0.978 |
| F1 Score | 0.973 | 0.979 | 0.857 |
| Precision | 0.972 | 0.972 | 0.762 |

Table 1 - Results

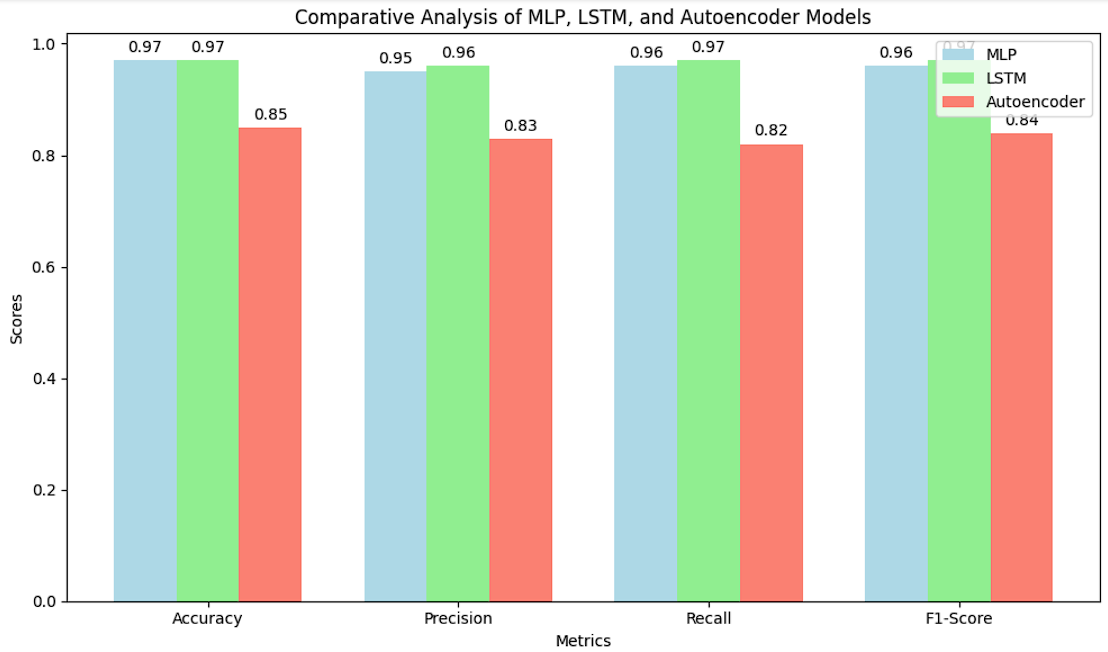


Fig. 11 Comparative analysis of Deep Learning models

**Comparison Summary:**

The graph compares the performance of MLP, LSTM, and Autoencoder across four metrics (Accuracy, Precision, Recall, and F1-Score). Both MLP and LSTM achieved high scores across all metrics, with LSTM slightly outperforming MLP in recall and F1-score, making it effective for sequential data. The Autoencoder, while lower in classification metrics, showed potential in unsupervised anomaly detection, making it useful for scenarios without predefined labels.

**VI . CONCLUSION**

This study presents a comprehensive comparative analysis of MLP, LSTM, and Autoencoder models for network intrusion detection, using the NSL-KDD dataset. The MLP and LSTM models demonstrated high accuracy and robustness in identifying diverse attack types, with LSTM proving particularly effective in capturing temporal dependencies within sequential data. While Autoencoder achieved slightly lower classification accuracy, its unsupervised structure highlights its potential for anomaly detection without the need for predefined labels, making it valuable for detecting unknown or evolving attack patterns. These findings suggest that integrating deep learning models into NIDS can provide robust detection capabilities, enhancing real-world network security. Future work could explore the development of hybrid models or ensemble methods to combine the unique strengths of MLP, LSTM, and Autoencoder, potentially improving both accuracy and adaptability in intrusion detection systems, especially in real-time or zero-day attack scenarios.

# References

1. Javaid, A., Niyaz, Q., Sun, W., & Alam, M. (2016, May). A deep learning approach for network intrusion detection system. In Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS) (pp. 21-26).
2. Jahan, F., Sun, W., Niyaz, Q., & Alam, M. (2019). Security modeling of autonomous systems: A survey. ACM Computing Surveys (CSUR), 52(5), 1-34.
3. Maithem, M., & Al-Sultany, G. A. (2021, February). Network intrusion detection system using deep neural networks. In Journal of Physics: Conference Series (Vol. 1804, No. 1, p. 012138). IOP Publishing.
4. Ashiku, L., & Dagli, C. (2021). Network intrusion detection system using deep learning. Procedia Computer Science, 185, 239-247.
5. Al-Emadi, S., Al-Mohannadi, A., & Al-Senaid, F. (2020, February). Using deep learning techniques for network intrusion detection. In 2020 IEEE international conference on informatics, IoT, and enabling technologies (ICIoT) (pp. 171-176). IEEE.
6. Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. IEEE transactions on emerging topics in computational intelligence, 2(1), 41-50.
7. Dinh, P. V., Ngoc, T. N., Shone, N., MacDermott, Á., & Shi, Q. (2017, November). Deep learning combined with de-noising data for network intrusion detection. In *2017 21st Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES)* (pp. 55-60). IEEE.
8. Seraphim, B. I., Palit, S., Srivastava, K., & Poovammal, E. (2018, December). A survey on machine learning techniques in network intrusion detection system. In *2018 4th International Conference on Computing Communication and Automation (ICCCA)* (pp. 1-5). IEEE.
9. Slevi, S. T., & Visalakshi, P. (2021, November). A survey on deep learning based intrusion detection systems on Internet of Things. In *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)* (pp. 1488-1496). IEEE.
10. Shende, S., & Thorat, S. (2020). Long short-term memory (LSTM) deep learning method for intrusion detection in network security. *International Journal of Engineering Research and*, *9*(06).
11. Kasongo, S. M. (2023). A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework. *Computer Communications*, *199*, 113-125.
12. Kasongo, S. M., & Sun, Y. (2019). A deep learning method with filter based feature engineering for wireless intrusion detection system. *IEEE access*, *7*, 38597-38607.
13. Kasongo, S. M., & Sun, Y. (2021). A deep gated recurrent unit based model for wireless intrusion detection system. *ICT Express*, *7*(1), 81-87.
14. Aldhaheri, A., Alwahedi, F., Ferrag, M. A., & Battah, A. (2024). Deep learning for cyber threat detection in IoT networks: A review. *Internet of Things and cyber-physical systems*, *4*, 110-128.
15. Alwahedi, F., Aldhaheri, A., Ferrag, M. A., Battah, A., & Tihanyi, N. (2024). Machine learning techniques for IoT security: Current research and future vision with generative AI and large language models. *Internet of Things and Cyber-Physical Systems*.
16. Fakhar, M., & Haile, A. (2022). AI for Threat Intelligence: Enhancing Adaptive Cyber Defense Against Persistent Attacks.
17. Ann, S., Cho, S. J., & Kim, H. (2024, July). A Preliminary Study on an Intrusion Detection Method using Large Language Models in Industrial Control Systems. In *2024 Fifteenth International Conference on Ubiquitous and Future Networks (ICUFN)* (pp. 600-602). IEEE.
18. Katiyar, N., Tripathi, M. S., Kumar, M. P., Verma, M. S., Sahu, A. K., & Saxena, S. (2024). AI and Cyber-Security: Enhancing threat detection and response with machine learning. *Educational Administration: Theory and Practice*, *30*(4), 6273-6282.
19. Byrapuneni, L. P., & SaidiReddy, M. (2024). An Advanced Filter-based Supervised Threat Detection Framework on Large Databases. *Engineering, Technology & Applied Science Research*, *14*(4), 15681-15685.
20. Byrapuneni, L. P., & Saidireddy, M. (2024). An Efficient Cluster Based Multi-Label Classification Model for Advanced Persistent Threat Attacks Detecting. *International Journal of Safety & Security Engineering*, *14*(2).
21. Aljuaid, W. A. H., & Alshamrani, S. S. (2024). A deep learning approach for intrusion detection systems in cloud computing environments. *Applied Sciences*, *14*(13), 5381.