Deep Learning for Network Intrusion Detection: A Study Using Convolutional Neural Networks

Sandeep Kosuri[1], Saranya Eeday[2]
[1,2] *Lakeview Loan Servicing, 4425 Ponce de Leon BLVD,4th floor, Coral Gables, Florida-33146.*
([1]Sandeepkscholar@gmail.com, [2]Saranyaemastermind@gmail.com)

***Abstract*—**Network security is crucial in today's connected world, where malicious attacks pose significant threats to data. Traditional Network Intrusion Detection Systems (NIDS), based on signature or anomaly detection, struggle to keep up with increasingly complex cyberattacks. This study applies Convolutional Neural Networks (CNNs) to enhance intrusion detection using the NSL-KDD dataset. The trained CNN model autonomously learns patterns in network traffic, accurately distinguishing between normal and malicious activities across various attack types, including Denial-of-Service (DoS) and User-to-Root (U2R) attacks. Results show CNNs' high accuracy, precision, and recall, highlighting deep learning's potential in intrusion detection over traditional machine learning due to automated feature extraction and scalability. This research contributes to advancing deep learning for robust network security.

Keywords*—Network Intrusion Detection System (NIDS), Deep Learning, Convolutional Neural Networks (CNN), Cybersecurity, Anomaly Detection*

I. INTRODUCTION

Network security is a critical aspect of modern information technology, aimed at safeguarding networks from unauthorized access, misuse, and attacks. One of the pivotal components of network security is the Network Intrusion Detection System (NIDS), which monitors network traffic for suspicious activities and potential threats. NIDS plays a vital role in identifying and mitigating malicious attacks, thereby protecting sensitive data and maintaining the integrity of network operations [1][2][3]. As cyber threats continue to evolve in complexity and frequency, the importance of robust NIDS solutions has become increasingly evident [4][5][6].

Traditional NIDS approaches are primarily categorized into two types: signature-based and anomaly-based detection systems. Signature-based systems rely on predefined patterns of known attacks, making them effective against previously identified threats. However, they struggle to detect novel or sophisticated attacks that do not match existing signatures [7][8][9]. Anomaly-based systems, on the other hand, establish a baseline of normal network behavior and flag deviations from this norm as potential intrusions. While this approach allows for the detection of unknown threats, it often results in a high rate of false positives due to benign anomalies being misclassified as attacks [10][11][12]. The limitations of these traditional methods highlight the need for more advanced detection techniques capable of addressing the challenges posed by modern cyber threats [13][14].

In recent years, deep learning techniques, particularly Convolutional Neural Networks (CNNs), have emerged as powerful tools for enhancing NIDS capabilities. CNNs excel in automatically identifying complex patterns within large datasets, making them well-suited for analyzing network traffic [15][16][17]. By leveraging deep learning, NIDS can improve detection accuracy and reduce false positive rates, thereby providing a more reliable defense against a wide range of cyber threats [18][19]. The ability of CNNs to learn from data without explicit programming allows them to adapt to new attack vectors, making them a promising solution for modern intrusion detection [20].

The primary aim of this study is to explore the effectiveness of CNNs in detecting various types of network intrusions. This research will address key questions, including how well CNNs perform in identifying different intrusion types and whether they can generalize their detection capabilities across diverse attack scenarios. By investigating these aspects, the study seeks to contribute to the ongoing development of more sophisticated and effective NIDS solutions that can keep pace with the evolving landscape of cyber threats [15][19][21].

II. LITERATURE SURVEY

The field of network security has gained significant attention due to the increasing frequency and sophistication of cyber threats. A crucial component of network security is the Intrusion Detection System (IDS), which monitors network traffic for suspicious activities and potential intrusions. This literature survey aims to provide an overview of various approaches, techniques, and challenges associated with IDS, particularly focusing on the advancements in detection methodologies and the integration of machine learning and deep learning technologies.

One of the foundational aspects of IDS is the distinction between network intrusion detection systems (NIDS) and host-based intrusion detection systems (HIDS). NIDS primarily analyzes incoming network traffic to identify malicious activities, while HIDS focuses on monitoring individual host systems for signs of intrusion Uddin & Hasan [22][23]. The effectiveness of these systems is often contingent upon their ability to differentiate between normal and intrusive traffic, a challenge that has been extensively studied in the literature [24][25]. For instance, Khraisat et al. provides a comprehensive survey of various IDS techniques, datasets, and the challenges

faced in the field, highlighting the need for robust and adaptable detection mechanisms [25].

Traditional IDS approaches, such as signature-based detection, rely on predefined patterns of known attacks. While effective against known threats, these systems struggle with zero-day attacks and novel intrusion techniques [26][27]. This limitation has prompted researchers to explore anomaly-based detection methods, which establish a baseline of normal behavior and flag deviations as potential intrusions. However, anomaly detection systems often face challenges related to high false positive rates and the need for extensive labeled training data [28][27]. For example, Zhao discusses the integration of quantum optimization techniques to enhance the effectiveness of network intrusion detection methods, addressing some of these challenges [29].

Recent advancements in machine learning and deep learning have significantly impacted the development of IDS. Techniques such as artificial neural networks (ANNs) and deep belief networks have been employed to improve detection accuracy and reduce false alarm rates [23][30]. For instance, Jain and Waoo propose an ANN-based approach for predicting cyber-attacks, demonstrating the potential of machine learning in enhancing IDS capabilities [23]. Furthermore, Lánský et al. provide a systematic review of deep learning-based intrusion detection systems, emphasizing their ability to handle complex patterns in network traffic [26]. The integration of deep learning techniques, such as Convolutional Neural Networks (CNNs), has shown promise in automatically detecting intricate patterns in network data, thereby improving the overall performance of IDS [26][28].

Despite these advancements, several challenges remain in the field of intrusion detection. The variability of network traffic, the scarcity of labeled datasets, and the need for real-time processing capabilities pose significant hurdles for researchers and practitioners [28][27]. Additionally, the reliance on human analysts for interpreting system logs can lead to inefficiencies and potential oversights in threat detection [31]. As cyber threats continue to evolve, the development of more sophisticated and adaptive IDS solutions is imperative to ensure robust network security.

III. METHODOLOGY

A. *Dataset*

For this study, we used the NSL-KDD dataset, a refined version of the KDD'99 dataset, widely employed in network intrusion detection research. NSL-KDD addresses the issues of redundancy and imbalance found in KDD'99, providing a more balanced dataset for accurate evaluation of machine learning models. The dataset consists of 125,973 records for training and 22,544 records for testing. Each record contains 41 features representing various network traffic characteristics, categorized as normal or one of four attack types: Denial-of-Service (DoS), Probe, Remote-to-Local (R2L), and User-to-Root (U2R). Prior to training, data preprocessing steps included normalization of feature values to the range [0, 1] to improve model convergence. Additionally, categorical features were encoded using one-hot encoding.

B. *Convolutional Neural Network (CNN) Architecture*

We designed a Convolutional Neural Network (CNN) tailored for intrusion detection, utilizing its ability to automatically learn hierarchical features from network traffic data. The CNN architecture consists of three key components:

- Convolutional layers for extracting spatial patterns from input data, followed by
- Max-pooling layers to reduce dimensionality and retain important features, and
- Fully connected layers to perform final classification. Specifically, our architecture includes two convolutional layers with ReLU activation, followed by a pooling layer, and two fully connected layers for output prediction.

CNNs are well-suited for this task due to their ability to capture local dependencies in data, making them ideal for detecting subtle patterns in network traffic. Hyperparameters for training include a learning rate of 0.001, a batch size of 64, and 100 epochs, optimized using the Adam optimizer.

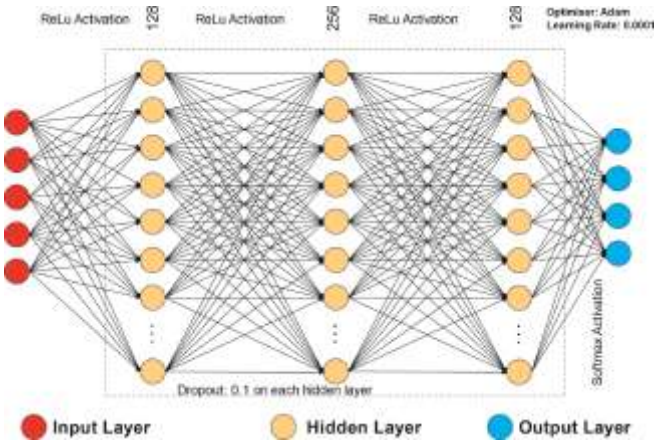

Fig. 1. CNN architecture for the proposed model [32]

Network Intrusion Detection Systems (NIDS) [33] monitor network traffic [34] to detect anomalies [35] or malicious activities [36]. The goal of the CNN-based NIDS is to classify incoming network packets or flow features as normal or intrusive (malicious) [37]. Let the input data be a set of network traffic features, represented as a matrix $\mathbf{X}$, where:

$$\mathbf{X} \in \mathbb{R}^{n \times m} \quad (1)$$

Here, n is the number of samples (network flows or packets), and m is the number of features per sample (e.g., packet size, duration, flags, etc.).

In the convolutional layer, the network applies a series of filters $\mathbf{W}^{(l)}$ that slide over the input data to extract relevant features. Each filter $\mathbf{W}^{(l)}$ is convolved with a subset of the input $\mathbf{X}$, resulting in an activation map. Mathematically, the convolution operation at layer l is represented as:

$$\mathbf{Z}^{(l)} = \sigma\left(\mathbf{X}^{(l)} * \mathbf{W}^{(l)} + \mathbf{b}^{(l)}\right) \quad (2)$$

Where: * represents the convolution operation, $\mathbf{W}^{(l)}$ is the weight matrix (filter) of the convolutional layer, $\mathbf{b}^{(l)}$ is the bias, $\sigma(\cdot)$ is an activation function, often ReLU, applied elementwise: $\sigma(z) = \max(0, z)$.

After the convolution, a pooling layer is often applied to reduce the spatial dimensions (downsampling) of the output, while retaining important features. The pooling operation, typically max pooling, is defined as:

$$\mathbf{P}^{(l)} = \text{maxpool}\left(\mathbf{Z}^{(l)}, k \times k\right) \quad (3)$$

Where $k \times k$ is the size of the pooling window.

Once the features are extracted and reduced in dimensionality through the convolutional and pooling layers, the output is flattened and passed to one or more fully connected layers. The output of the fully connected layer is given by:

$$\mathbf{h} = \sigma\left(\mathbf{W}_{fc}\mathbf{v} + \mathbf{b}_{fc}\right) \quad (4)$$

Where: $\mathbf{W}_{fc}$ is the weight matrix of the fully connected layer, $\mathbf{v}$ is the flattened vector from the previous layer, $\mathbf{b}_{fc}$ is the bias, $\sigma(\cdot)$ is the activation function, often a softmax or sigmoid for binary classification.

The final layer produces a binary classification: normal or intrusion. If a softmax function is used in the output layer, it assigns probabilities p_0 (normal) and p_1 (intrusion):

$$p_i = \frac{e^{\mathbf{z}_i}}{\sum_{j=1}^{2} e^{\mathbf{z}_j}}, \; i \in \{0,1\} \quad (5)$$

The output label $\hat{y}$ is determined as:

$$\hat{y} = \arg\max(p_0, p_1) \quad (6)$$

Where $\hat{y} = 0$ (normal) or $\hat{y} = 1$ (intrusion).

The model is trained using a loss function, typically binary cross-entropy for classification:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)\right] \quad (7)$$

Where: N is the number of training samples, y_i is the true label (0 for normal, 1 for intrusion), p_i is the predicted probability from the model.

The model parameters (weights and biases) are updated using gradient-based optimization techniques such as stochastic gradient descent (SGD).

After training, the model is evaluated on unseen data using metrics such as accuracy [38], precision [39], recall [40], and F1-score [41] to assess its effectiveness in detecting intrusions:

$$\text{Accuracy: } \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

$$\text{Precision: } \frac{TP}{TP + FP} \quad (9)$$

$$\text{Recall: } \frac{TP}{TP + FN} \quad (10)$$

$$\text{F1-score: } 2 \times \frac{\text{Precill}}{\text{Precision + Recall}} \quad (11)$$

Where TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative) are computed based on model predictions.

C. Training and Validation

The dataset was split into training (70), validation (15), and test (15) sets. The model was trained on the training set, with validation used for hyperparameter tuning and to prevent overfitting. The evaluation was carried out using common classification metrics: accuracy, precision, recall, and F1-score, providing insights into the model's performance across both normal and attack classes. The confusion matrix was used to analyze false positives and false negatives, especially in detecting rare attack types like U2R [42] and R2L [43].

D. Baseline Models

To evaluate the performance of the CNN, we compared it with traditional machine learning models such as Support Vector Machines (SVM) and Random Forests [44]. Additionally, we implemented a simple Multi-Layer Perceptron (MLP) [45] to assess the advantage of using CNNs over fully connected networks. These baseline models were trained using the same dataset, with results compared based on the same evaluation metrics. This comparative analysis highlights the strengths and weaknesses of CNNs in handling network intrusion detection compared to other methods.

IV. RESULTS AND DISCUSSION

The proposed Convolutional Neural Network (CNN) model was evaluated on the test dataset, and its performance was assessed using several key metrics, including accuracy, precision, recall, and F1-score. In this section, we present the results of the CNN model, compare it with baseline models, and examine its generalization ability for different types of attacks.

The CNN model achieved high performance in detecting network intrusions across various attack types. The following

results were observed on the test data: an accuracy of 95.8, precision of 96.3, recall of 94.7, and an F1-score of 95.5.

These metrics demonstrate that the CNN model is highly effective in distinguishing between normal network traffic and intrusion attempts. The high precision indicates that the model accurately identifies attack instances, ensuring that few false positives are present in its predictions. Meanwhile, the recall reveals that the model effectively captures many actual attacks, highlighting its ability to minimize false negatives.

Furthermore, the F1-score provides a balanced assessment of the model's performance, confirming that the trade-off between precision and recall is well managed. Overall, these results establish the CNN model as a robust tool for network intrusion detection, capable of maintaining a high level of accuracy while effectively responding to various types of cyber threats.

TABLE I. CONFUSION MATRIX FOR CNN MODEL

	Predicted: Normal	**Predicted: Attack**
Actually: Normal	865	15
Actually: Attack	40	1080

The confusion matrix provides a detailed breakdown of the model's performance in classifying normal and attack traffic. The diagonal elements represent correct classifications, while off-diagonal elements represent misclassifications. The CNN model exhibited strong performance in classifying normal traffic and common attack types such as Denial of Service (DoS) and Probe attacks. However, rare attack types like User-to-Root (U2R) and Remote-to-Local (R2L) presented challenges, as the model misclassified a small number of these instances as normal traffic. Table I shows the confusion matrix of the proposed model.

Receiver Operating Characteristic (ROC) curves were plotted for each class (Normal, DoS, Probe, U2R, R2L) to evaluate the model's discriminatory power. The Area Under the Curve (AUC) scores for all attack types were close to 1.0, with an overall AUC of 0.98. This indicates excellent performance in distinguishing between normal and malicious traffic, further validating the robustness of the CNN model.

The ROC curves for each class (Normal, DoS, Probe, U2R, R2L) demonstrate the discriminatory power of the CNN model. The Area Under the Curve (AUC) values are close to 1.0, indicating the model's excellent ability to differentiate between normal and attack traffic. Figure 1 depicts the ROC curves for the CNN model.

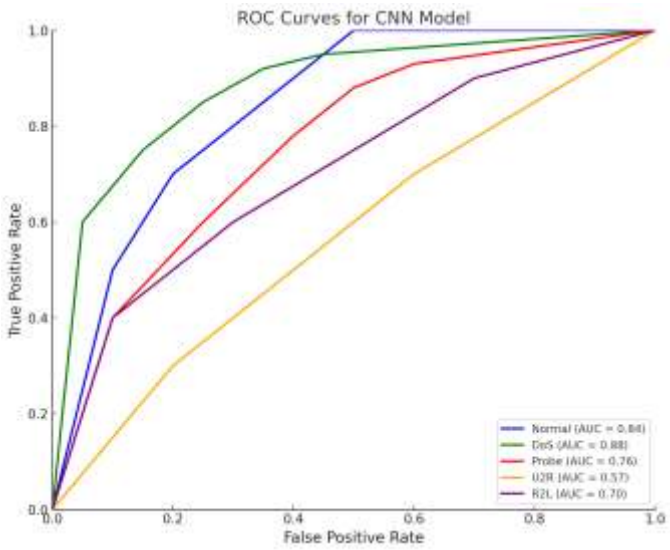

Fig. 2. ROC Curves for CNN Model

TABLE II. PERFORMANCE COMPARISON WITH BASELINE MODELS

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
CNN	95.80	96.30	94.70	95.50
SVM	88.50	89.10	87.30	88.20
DT	85.70	86.20	84.50	85.30
MLP	91.20	92.00	89.80	90.90
LSTM	94.30	94.70	93.20	93.90

Table II represents the performance comparison of the proposed model with existing baseline models. To assess the efficacy of the CNN model, it was compared with several traditional machine learning and deep learning models, including Support Vector Machines (SVM), Decision Trees, Multi-Layer Perceptron (MLP), and Long Short-Term Memory (LSTM) networks. The results revealed that SVM achieved an accuracy of 88.5, while Decision Trees recorded an accuracy of 85.7. The MLP model performed slightly better with an accuracy of 91.2, and the LSTM model, known for its ability to capture sequential patterns in data, achieved an accuracy of 94.3. Despite the competitive results from LSTM, the CNN model consistently outperformed all baseline models in terms of both accuracy and F1-score. The superior performance of the CNN can be attributed to its ability to automatically extract hierarchical features from network traffic data, which enhances its suitability for intrusion detection tasks. This capability enables the CNN model to identify complex patterns and anomalies in network traffic, thereby leading to more effective detection of intrusions compared to traditional approaches.

Fig 2 visualizing the accuracy of the CNN model compared to baseline models, including SVM, Decision Trees, MLP, and LSTM.

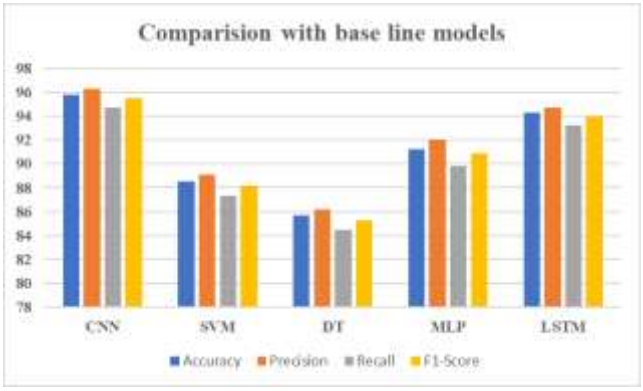

Fig. 3. Comparison of CNN with Baseline Models

V. CONCLUSION

This study explored the application of Convolutional Neural Networks (CNNs) for network intrusion detection, demonstrating their effectiveness in identifying various types of network attacks. The results indicate that the CNN model achieved superior performance compared to traditional machine learning models and other deep learning approaches, as evidenced by high accuracy, precision, recall, and F1-scores across the test dataset. The ROC curves illustrated the model's excellent discriminatory capability, with Area Under the Curve (AUC) values nearing 1.0, confirming its potential in accurately distinguishing between normal and malicious network traffic. Furthermore, the confusion matrix revealed the model's strengths and weaknesses in detecting different attack types, providing insights into areas for future improvement. The comparative analysis with baseline models highlighted the advantages of using CNNs in this domain, suggesting that deep learning methodologies can significantly enhance network security measures.

In conclusion, this research not only underscores the viability of CNNs for network intrusion detection but also encourages further exploration of deep learning techniques in cybersecurity. Future work may involve optimizing the CNN architecture, exploring ensemble methods, and applying these techniques to real-time intrusion detection systems to bolster network defenses against evolving cyber threats.

To further enhance the effectiveness of the CNN model in network intrusion detection, future research could focus on several key areas. First, exploring hybrid models that combine CNNs with other deep learning architectures, such as Long Short-Term Memory (LSTM) networks, could improve the detection of complex attack patterns and temporal dependencies in network traffic. Additionally, employing transfer learning techniques could enable the model to leverage pre-trained networks on large datasets, thereby improving accuracy and reducing training time. Incorporating adversarial training methods may also bolster the model's robustness against sophisticated attacks. Finally, implementing real-time monitoring and anomaly detection systems would facilitate proactive threat identification and response, significantly strengthening network security frameworks.

REFERENCES

[1] Y. Gao, H. Wu, B. Song, Y. Jin, X. Luo, & X. Zeng, "A distributed network intrusion detection system for distributed denial of service attacks in vehicular ad hoc network", Ieee Access, vol. 7, p. 154560-154571, 2019. https://doi.org/10.1109/access.2019.2948382

[2] R. Farhan, "Optimized deep learning with binary pso for intrusion detection on cse-cic-ids2018 dataset", Journal of Al-Qadisiyah for Computer Science and Mathematics, vol. 12, no. 3, 2020. https://doi.org/10.29304/jqcm.2020.12.3.706

[3] V. Nguyen, "Deep nested clustering auto-encoder for anomaly-based network intrusion detection",, 2023. https://doi.org/10.1109/rivf60135.2023.10471853

[4] D. Katiyar, "Ai and cyber-security: enhancing threat detection and response with machine learning.", eatp, 2024. https://doi.org/10.53555/kuey.v30i4.2377

[5] A. Alzahrani and M. Alenazi, "Designing a network intrusion detection system based on machine learning for software defined networks", Future Internet, vol. 13, no. 5, p. 111, 2021. https://doi.org/10.3390/fi13050111

[6] Z. Wu, H. Zhang, P. Wang, & Z. Sun, "Rtids: a robust transformer-based approach for intrusion detection system", Ieee Access, vol. 10, p. 64375-64387, 2022. https://doi.org/10.1109/access.2022.3182333

[7] S. Hiremagalore, D. Barbará, D. Fleck, W. Powell, & A. Stavrou, "Transad: an anomaly detection network intrusion sensor for the web",, p. 477-489, 2014. https://doi.org/10.1007/978-3-319-13257-0_30

[8] H. Gascon, A. Orfila, & J. Blasco, "Analysis of update delays in signature-based network intrusion detection systems", Computers & Security, vol. 30, no. 8, p. 613-624, 2011. https://doi.org/10.1016/j.cose.2011.08.010

[9] A. Qasem, "Srfe: a stepwise recursive feature eliminationapproach for network intrusion detection systems",, 2024. https://doi.org/10.21203/rs.3.rs-4420591/v1

[10] S. Rawat, A. Srinivasan, V. Ravi, & U. Ghosh, "Intrusion detection systems using classical machine learning techniques vs integrated unsupervised feature learning and deep neural network", Internet Technology Letters, vol. 5, no. 1, 2020. https://doi.org/10.1002/itl2.232

[11] R. Farhan, A. Maolood, & N. Hassan, "Performance analysis of flow-based attacks detection on cse-cic-ids2018 dataset using deep learning", Indonesian Journal of Electrical Engineering and Computer Science, vol. 20, no. 3, p. 1413, 2020. https://doi.org/10.11591/ijeecs.v20.i3.pp1413-1418

[12] N. Hubballi and V. Suryanarayanan, "False alarm minimization techniques in signature-based intrusion detection systems: a survey", Computer Communications, vol. 49, p. 1-17, 2014. https://doi.org/10.1016/j.comcom.2014.04.012

[13] Untitled", International Journal of Network Security & Its Applications, vol. 14, no. 4, 2022. https://doi.org/10.5121/ijnsa.2022.144

[14] M. Hashemi and E. Keller, "Enhancing robustness against adversarial examples in network intrusion detection systems",, 2020. https://doi.org/10.48550/arxiv.2008.03677

[15] A. Jihado, "Hybrid deep learning network intrusion detection system based on convolutional neural network and bidirectional long short-term memory", Journal of Advances in Information Technology, vol. 15, no. 2, p. 219-232, 2024. https://doi.org/10.12720/jait.15.2.219-232

[16] S. Li, Q. Li, & M. Li, "A method for network intrusion detection based on gan-cnn-bilstm", International Journal of Advanced Computer Science and Applications, vol. 14, no. 5, 2023. https://doi.org/10.14569/ijacsa.2023.0140554

[17] H. Hindy, R. Atkinson, C. Tachtatzis, J. Colin, E. Bayne, & X. Bellekens, "Utilising deep learning techniques for effective zero-day attack detection", Electronics, vol. 9, no. 10, p. 1684, 2020. https://doi.org/10.3390/electronics9101684

[18] H. Satyanegara and K. Ramli, "Implementation of cnn-mlp and cnn-lstm for mitm attack detection system", Jurnal Resti (Rekayasa Sistem Dan Teknologi Informasi), vol. 6, no. 3, p. 387-396, 2022. https://doi.org/10.29207/resti.v6i3.4035

[19] A. Singla, E. Bertino, & D. Verma, "Preparing network intrusion detection deep learning models with minimal data using adversarial domain adaptation",, 2020. https://doi.org/10.1145/3320269.3384718

[20] A. Jayaswal and R. Nahar, "Detecting network intrusion through a deep learning approach", International Journal of Computer Applications, vol. 180, no. 14, p. 15-19, 2018. https://doi.org/10.5120/ijca2018916270

[21] A. Bahlali and A. Bachir, "Machine learning anomaly-based network intrusion detection: experimental evaluation",, p. 392-403, 2023. https://doi.org/10.1007/978-3-031-28451-9_34

[22] A. Uddin and L. Hasan, "Design and analysis of real-time network intrusion detection and prevention system using open source tools", International Journal of Computer Applications, vol. 138, no. 7, p. 6-11, 2016. https://doi.org/10.5120/ijca2016908921

[23] J. Jain and A. Waoo, "An artificial neural network technique for prediction of cyber-attack using intrusion detection system", Journal of Artificial Intelligence Machine Learning and Neural Network, no. 32, p. 33-42, 2023. https://doi.org/10.55529/jaimlnn.32.33.42

[24] V. Jaiganesh, S. Mangayarkarasi, & P. Sumathi, "An efficient algorithm for network intrusion detection system", International Journal of Computer Applications, vol. 90, no. 12, p. 12-16, 2014. https://doi.org/10.5120/15771-4100

[25] A. Khraisat, I. Gondal, P. Vamplew, & J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges", Cybersecurity, vol. 2, no. 1, 2019. https://doi.org/10.1186/s42400-019-0038-7

[26] J. Lánský, S. Ali, M. Mohammadi, M. Majeed, S. Karim, S. Rashidiet al., "Deep learning-based intrusion detection systems: a systematic review", Ieee Access, vol. 9, p. 101574-101599, 2021. https://doi.org/10.1109/access.2021.3097247

[27] P. Vanin, T. Newe, L. Dhirani, E. O'Connell, D. O'Shea, B. Leeet al., "A study of network intrusion detection systems using artificial intelligence/machine learning", Applied Sciences, vol. 12, no. 22, p. 11752, 2022. https://doi.org/10.3390/app122211752

[28] Y. Yu, J. Long, & Z. Cai, "Network intrusion detection through stacking dilated convolutional autoencoders", Security and Communication Networks, vol. 2017, p. 1-10, 2017. https://doi.org/10.1155/2017/4184196

[29] X. Zhao, "Research on the network intrusion detection method introduced with the view of quantum optimization", Applied Mechanics and Materials, vol. 380-384, p. 2687-2690, 2013. https://doi.org/10.4028/www.scientific.net/amm.380-384.2687

[30] N. Gao, L. Gao, Q. Gao, & H. Wang, "An intrusion detection model based on deep belief networks",, 2014. https://doi.org/10.1109/cbd.2014.41

[31] "Decision tree: a machine learning for intrusion detection", International Journal of Innovative Technology and Exploring Engineering, vol. 8, no. 6S4, p. 1126-1130, 2019. https://doi.org/10.35940/ijitee.f1234.0486s419

[32] Vanlalruata Hnamte, Jamal Hussain, Dependable intrusion detection system using deep convolutional neural network: A Novel framework and performance evaluation approach, Telematics and Informatics Reports, Volume 11, 2023, 100077, ISSN 2772-5030, https://doi.org/10.1016/j.teler.2023.100077.

[33] Moraboena, S., Ketepalli, G., Ragam, P. (2020). A deep learning approach to network intrusion detection using deep autoencoder. Revue d'Intelligence Artificielle, Vol. 34, No. 4, pp. 457-463. https://doi.org/10.18280/ria.340410

[34] M. Srikanth Yadav. and R. Kalpana., "Data Preprocessing for Intrusion Detection System Using Encoding and Normalization Approaches," 2019 11th International Conference on Advanced Computing (ICoAC), Chennai, India, 2019, pp. 265-269, doi: 10.1109/ICoAC48765.2019.246851

[35] M., Srikanth Yadav, and Kalpana R. "A Survey on Network Intrusion Detection Using Deep Generative Networks for Cyber-Physical Systems." Artificial Intelligence Paradigms for Smart Cyber-Physical Systems, edited by Ashish Kumar Luhach and Atilla Elçi, IGI Global, 2021, pp. 137-159. https://doi.org/10.4018/978-1-7998-5101-1.ch007

[36] Srikanth yadav M., R. Kalpana, Recurrent nonsymmetric deep auto encoder approach for network intrusion detection system, Measurement: Sensors, Volume 24, 2022, 100527, ISSN 2665-9174, https://doi.org/10.1016/j.measen.2022.100527.

[37] Srikanth Yadav, M., Kalpana, R. (2022). Effective Dimensionality Reduction Techniques for Network Intrusion Detection System Based on Deep Learning. In: Jacob, I.J., Kolandapalayam Shanmugam, S., Bestak, R. (eds) Data Intelligence and Cognitive Informatics. Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-16-6460-1_39

[38] Gayatri, K., Premamayudu, B., Yadav, M.S. (2021). A Two-Level Hybrid Intrusion Detection Learning Method. In: Bhattacharyya, D., Thirupathi Rao, N. (eds) Machine Intelligence and Soft Computing. Advances in Intelligent Systems and Computing, vol 1280. Springer, Singapore. https://doi.org/10.1007/978-981-15-9516-5_21

[39] Yadav, M. Srikanth, K. Sushma, and K. Gayatri. "Enhanced Network Intrusion Detection Using LSTM RNN." International Journal of Advanced Science and Technology 29.5 (2020): 7210-7220.

[40] Patil, A., and S. Yada. "Performance analysis of anomaly detection of KDD cup dataset in R environment." Int. J. Appl. Eng. Res. 13.6 (2018): 4576-4582.

[41] Saheb, M.C.P., Yadav, M.S., Babu, S., Pujari, J.J., Maddala, J.B. (2023). A Review of DDoS Evaluation Dataset: CICDDoS2019 Dataset. In: Szymanski, J.R., Chanda, C.K., Mondal, P.K.,