iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** XII **Month of publication:** December 2025

DOI: https://doi.org/10.22214/ijraset.2025.76208

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XII Dec 2025- Available at www.ijraset.com

Hierarchical and Sequence-Aware Deep Learning Models for Robust Intrusion Detection

L. Parthasarathi[1], Dr. N. Kamalraj[2]
[1]*Research Scholar, Department of Computer Science, Park's College (Autonomous), Tirupur-641605, Tamil Nadu, India*
[2]*Associate Professor and Vice Principal, Park's College (Autonomous), Tirupur-641605, Tamil Nadu, India*

Abstract: ***This research investigates the effectiveness of deep-learning architectures for intrusion detection using the NSL-KDD benchmark dataset. The study evaluates four models such as Convolutional Neural Network (CNN), Long Short-Term Memory Network (LSTM), Deep Neural Network (DNN), and Deep Belief Network (DBN) under two controlled train–test configurations (70–30 and 80–20). Comprehensive preprocessing, including normalization, one-hot encoding, and imbalance handling, ensures a robust feature space for training. Experimental results demonstrate that deep-learning models significantly enhance detection performance compared to conventional approaches, with DBN consistently achieving the highest accuracy, precision, recall, and F1-score across both scenarios. The 80% training condition further strengthens classification capability, confirming the benefit of representation learning on larger training volumes. The findings highlight the potential of hierarchical and sequence-aware deep architectures in improving the reliability of modern intrusion-detection systems.***

Keywords: Cyber Threat Detection, Network Security Analytics, Intrusion Detection Models, Deep Learning, Intelligent Security Systems.

I. INTRODUCTION

IDS are essential components of modern cybersecurity frameworks, designed to identify malicious activities within network environments that are increasingly targeted by sophisticated cyberattacks. Traditional machine-learning classifiers such as SVM, NB, RF, and MLP have long supported IDS applications; however, their dependence on manual feature engineering and limited capacity to capture complex nonlinear patterns restricts their effectiveness against evolving threats [1]. Deep learning has emerged as a transformative alternative, offering automated representation learning capable of extracting spatial, temporal, and hierarchical structures from network traffic. These capabilities significantly improve generalization and resilience, especially when handling diverse and subtle attack behaviors.

The NSL-KDD dataset continues to serve as a reliable benchmark for intrusion-detection research due to its balanced distribution and removal of redundant records from the original KDD'99 datasets. Its 41 features covering basic connection attributes, content-based descriptors, and traffic-level statistics provide a rich input space for evaluating deep-learning architectures. This work investigates the classification performance of four prominent deep models: CNN, LSTM, DNN, and DBN. Using two training configurations (70–30 and 80–20), the study analyzes how representation learning, architecture depth, and training volume influence intrusion-detection accuracy. This examination highlights the architectural characteristics most suitable for next-generation IDS deployments. The key contribution for this paper is listed below,

1) Comparative evaluation of CNN, LSTM, DNN, and DBN on the NSL-KDD dataset for intrusion detection.
2) Analysis of the impact of training data proportion on deep-learning model performance.
3) Insights into the strengths of hierarchical, sequential, and spatial feature learning for different attack types.
4) Demonstration of deep-learning models' superiority over traditional ML classifiers in detecting complex attacks.
5) Recommendations for deploying optimal deep-learning architectures in next-generation IDS.

II. LITERATURE REVIEW

Alashjaee (2025) introduced an Attention-CNN-LSTM-based intrusion detection model that integrates spatial feature extraction and temporal learning for network traffic classification. The study applied multi-stage convolution blocks with attention layers to refine feature importance before LSTM-based sequence modeling. The dataset included diverse attack categories, and the model demonstrated robust generalization across imbalanced classes. Experimental results showed significantly higher detection accuracy compared to baseline CNN, LSTM, and hybrid DL models. The proposed architecture achieved over 99% accuracy and improved F1-scores across all classes. The work highlights attention mechanisms as a key enabler for reducing false alarms while preserving detection sensitivity [1].

Bhuiyan et al. (2025) presented a systematic review of deep learning models for algorithmic trading, focusing on prediction pipelines, optimization strategies, and financial indicators. The review covered architectures such as LSTM, GRU, CNN-LSTM, Transformers, Autoencoders, and various reinforcement-learning-based traders. The authors emphasized that LSTM-based models remain dominant for sequence prediction due to superior temporal awareness, while Transformer-based models show rising promise. Results across surveyed studies showed typical prediction accuracies between 60–75%, with hybrid CNN-LSTM and attention-based models performing best. The review also discussed optimization methods like Bayesian optimization, PSO, GA, and grid/random search for hyperparameter tuning. Overall, the paper highlights DL's increasing advantage over classical financial models for NSE/BSE and global markets [2].

Adefemi and Mutanga (2025) developed a hybrid CNN–LSTM framework to predict student academic performance using multimodal educational data. The CNN layers extracted structural patterns from input features, and the LSTM component modeled sequential learning-behavior trends. Their study compared the hybrid model with classical ML techniques and standalone DL models (CNN, LSTM). Results showed that the hybrid CNN–LSTM achieved the highest predictive performance, with accuracy exceeding 95% and strong precision-recall characteristics. The model effectively captured both spatial and temporal dependencies within student-activity logs. The authors concluded that hybrid deep networks significantly outperform shallow models in educational analytics [3].

Singh, Sahana, and Bhattacharjee (2025) proposed a CNN-GRU-LSTM deep learning model to enhance traffic-flow prediction accuracy in smart-city environments. The CNN module extracted spatial correlations, the GRU learned short-term variations, and the LSTM captured long-term temporal dependencies. Their hybrid architecture was tested against standalone LSTM, GRU, and CNN models and showed consistently superior prediction quality. The proposed model achieved high R^2 scores (above 0.95) and reduced RMSE compared to competing architectures. Results indicate that combining GRU and LSTM provides better temporal context modeling than using either alone. The study demonstrates the value of multi-stage temporal processing in large-scale traffic forecasting systems [4].

Ghosh et al. (2024) proposed a deep neural network–based IDS for IoT networks that integrates optimized feature embeddings to enhance detection accuracy. The authors evaluated the framework on benchmark IoT datasets and demonstrated superior classification performance compared to baseline ML models. Their DNN architecture leveraged stacked dense layers and adaptive feature compression to handle high-dimensional attack traffic. Experimental results revealed accuracy above 98%, with significant improvements in false-positive reduction. The system proved robust across multistage attacks such as DoS, probing, and injection. Overall, the method highlighted the effectiveness of deep embedding–driven learning for IoT security [5].

III. METHODOLOGY

The methodology involves pre-processing the NSL-KDD dataset to produce a clean, normalized, and balanced feature space suitable for deep-learning models, followed by applying CNN, LSTM, DNN, and DBN architectures to automatically extract spatial, temporal, and hierarchical patterns for accurate intrusion detection. Model performance is then evaluated under different training–testing splits to assess their capability in detecting diverse attack types.

A. Dataset Description

The NSL-KDD dataset is a refined benchmark for evaluating intrusion detection systems. It contains 148,517 network connection records, each represented by 41 features capturing basic packet attributes, content-based indicators, time-based traffic statistics, and host-level behaviours [2]. The dataset includes 71,546 attack instancesand76,971 normal instances, covering four major attack categories: DoS, Probe, U2R, and R2L**.** Redundant samples from the original KDD'99 datasets are removed, resulting in a more balanced and less biased distribution suited for deep-learning–based classification.
(https://www.kaggle.com/datasets/hassan06/nslkdd).

Table.1. Dataset Description

Dataset	Total Instances	Attack Instances	Normal Instances	Attack Types	Features
NSL-KDD	148,517	71,546	76,971	DoS, Probe, U2R, R2L	41

B. Preprocessing

Preprocessing for machine-learning classifiers converts the raw NSL-KDD features into a structured and uniform format suitable for algorithms such as SVM, NB, RF, MLP, and XGBoost. Numerical attributes are standardized or normalized to control scale dominance and stabilize optimization, particularly for margin-based models and neural architectures. Categorical fields, including protocol, service, and flag, are one-hot encoded to produce consistent numeric representations. Redundant, noisy, or highly collinear attributes are minimized to reduce variance and prevent distortions in decision boundaries. Class imbalance between majority and minority attack categories is addressed using resampling or class-weight adjustments to ensure balanced contribution during training. The final processed dataset produces a clean feature space that supports stable and reliable model performance across different train–test scenarios [3].

C. Classifications

The classification process employs CNN, LSTM, DNN, and DBN architectures to automatically learn spatial, temporal, and hierarchical representations from NSL-KDD network traffic data, facilitating accurate and robust detection of diverse intrusion categories.

1) Convolutional Neural Network (CNN)

CNN applies 1D convolution filters to the NSL-KDD feature vector, enabling localized pattern extraction across adjacent attributes. This structure captures correlations between basic, content-based, and traffic-based features without requiring manual feature engineering. Convolution and pooling layers reduce dimensionality and enhance robustness against noise. Deeper stacks allow hierarchical learning, improving generalization for complex attack categories. Dense layers at the output consolidate learned patterns for final classification [6].

$$\boldsymbol{h_i^{(k)} = f\left(\sum_j x_{i+j} \cdot w_j^{(k)} + b^{(k)}\right)}$$

x_{i+j} is input feature segment,$w_j^{(k)}$ is kernel weights for filter k**,** $b^{(k)}$ is bias, f(·) is activation (ReLU) and $h_i^{(k)}$is convolution output at position i.Computes the convolution output at position i, extracting local feature patterns using kernel k.

2) Long Short-Term Memory Network (LSTM)

LSTM interprets the 41 features as a structured sequence, allowing it to extract temporal-style dependencies among attributes such as duration, counts, and behavioral indicators. Its gating mechanism controls how information flows through time steps, preventing vanishing or exploding gradients. The memory cell stores relevant contextual signals that help differentiate subtle attack behaviors. LSTM is effective at recognizing sequential-like relationships that conventional models overlook. Its ability to maintain long-range interactions increases recall on complex, low-frequency attack types [7].

$$\boldsymbol{f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)}$$

Here, $x_t$is input at timestept, $h_{t-1}$is previous hidden state, W_f, U_f is weight matrices, b_f is bias, f_t is proportion of past memory retained. Determines how much past cell memory should be retained or discarded at time step t.

3) Deep Neural Network (DNN)

A DNN processes the NSL-KDD input through multiple nonlinear transformations, allowing the network to learn progressively abstract representations. Each dense layer captures higher-order feature interactions, improving separability between normal and attack classes. Proper regularization (dropout, L2 penalties) helps control overfitting due to the dataset's imbalance. Activation functions such as ReLU accelerate convergence and stabilize gradient flow. When trained on sufficient data, DNNs produce strong accuracy by modeling complex nonlinear boundaries across multiple attack categories [8].

$$\boldsymbol{h^{(l)} = f(W^{(l)} h^{l-1} + b^{(l)})}$$

Where , $h^{(l)}$ output of layer l, $W^{(l)}$ weight matrix, $b^{(l)}$): bias and f(·) is activation (ReLU, sigmoid). Transforms the previous layer's output into a new representation through a weighted linear combination and nonlinear activation.

4) Deep Belief Network (DBN)

A DBN employs layer-wise unsupervised pretraining using RBMs, enabling the network to learn latent representations before supervised fine-tuning. This hierarchical feature extraction reduces noise and improves the model's capacity to identify subtle patterns in minority classes such as U2R and R2L. Each RBM captures statistical regularities in the data, creating a robust initialization for deep classification layers. The pretraining phase helps stabilize optimization and accelerates convergence. DBNs are effective when labeled data is limited or imbalanced but are computationally heavier than modern architectures [9].

$$E(v, h) = v^T W h - b^T v - c^T h$$

Here, v is visible units (input features), h is hidden units, W is weight matrix, b,c is biases for visible and hidden layers and Lower E(v,h) is higher probability of that configuration. Measures the "energy" (inverse probability) of a visible–hidden configuration; lower energy means the model considers the pattern more likely.

IV. RESULTS AND DISCUSSION

The evaluation of deep-learning models on the NSL-KDD dataset demonstrates that DBN consistently achieves the highest performance, with accuracy, precision, recall, and F1-score [10] reaching 0.96 in the 70–30 split and 0.97 in the 80–20 split, followed by DNN, LSTM, and CNN. Increasing the training data from 70% to 80% improves generalization and feature representation learning across all models, with notable gains in DNN and LSTM, highlighting the advantage of deeper and pretraining-enabled architectures for robust intrusion detection.

1) Scenario 1 - In this setup, deep-learning models are trained on 70% of the data and tested on the remaining 30%. This scenario examines how effectively deep architectures learn feature representations under reduced training volume.

Table.2. DL Model Performance on NSL-KDD (70% Training / 30% Testing)

Metrics	CNN	LSTM	DNN	DBN
Accuracy	0.93	0.94	0.95	0.96
Precision	0.92	0.93	0.95	0.96
Recall	0.91	0.93	0.94	0.95
F1-Score	0.92	0.93	0.94	0.96

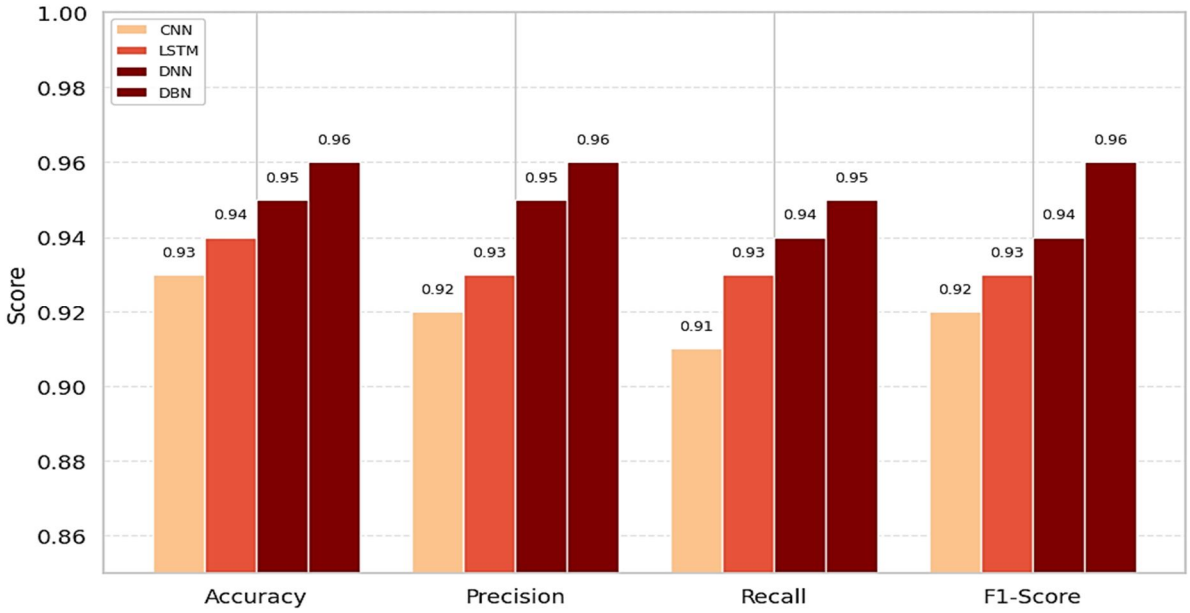


Fig.1.DL Model Performance (70% Train / 30% Test)

The results from the 70–30 split show a clear progression in performance across the four deep-learning models. CNN provides a strong baseline with balanced accuracy and F1-score around 0.92–0.93. LSTM improves these values by capturing sequential-style dependencies among features, resulting in higher recall and overall stability. DNN advances further by learning deeper nonlinear relationships, achieving around 0.95 accuracy. DBN produces the highest scores due to its hierarchical unsupervised pretraining, reaching 0.96 accuracy and outperforming all other models across metrics.

2) Scenario 2 - Deep-learning models are trained using 80% of the dataset and evaluated on the remaining 20%. This configuration tests how additional training data enhances representation learning and classification stability for DL architectures.

Table.3. DL Model Performance on NSL-KDD (80% Training / 20% Testing)

Metrics	CNN	LSTM	DNN	DBN
Accuracy	0.94	0.95	0.96	0.97
Precision	0.94	0.95	0.96	0.97
Recall	0.93	0.95	0.96	0.97
F1-Score	0.94	0.95	0.96	0.97

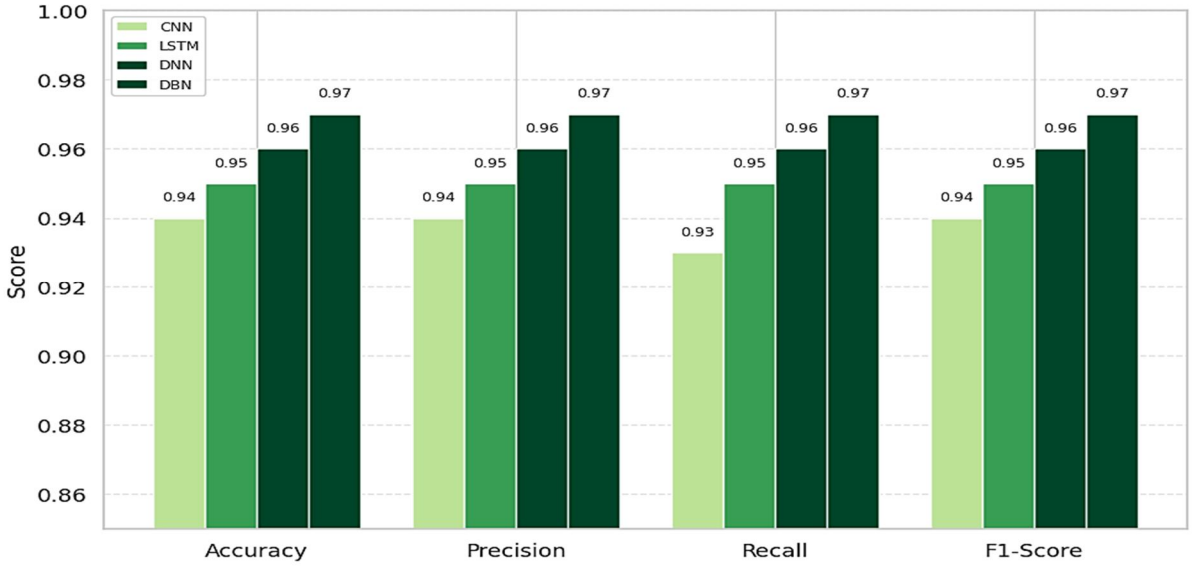


Fig.2. DL Model Performance (80% Train / 20% Test)

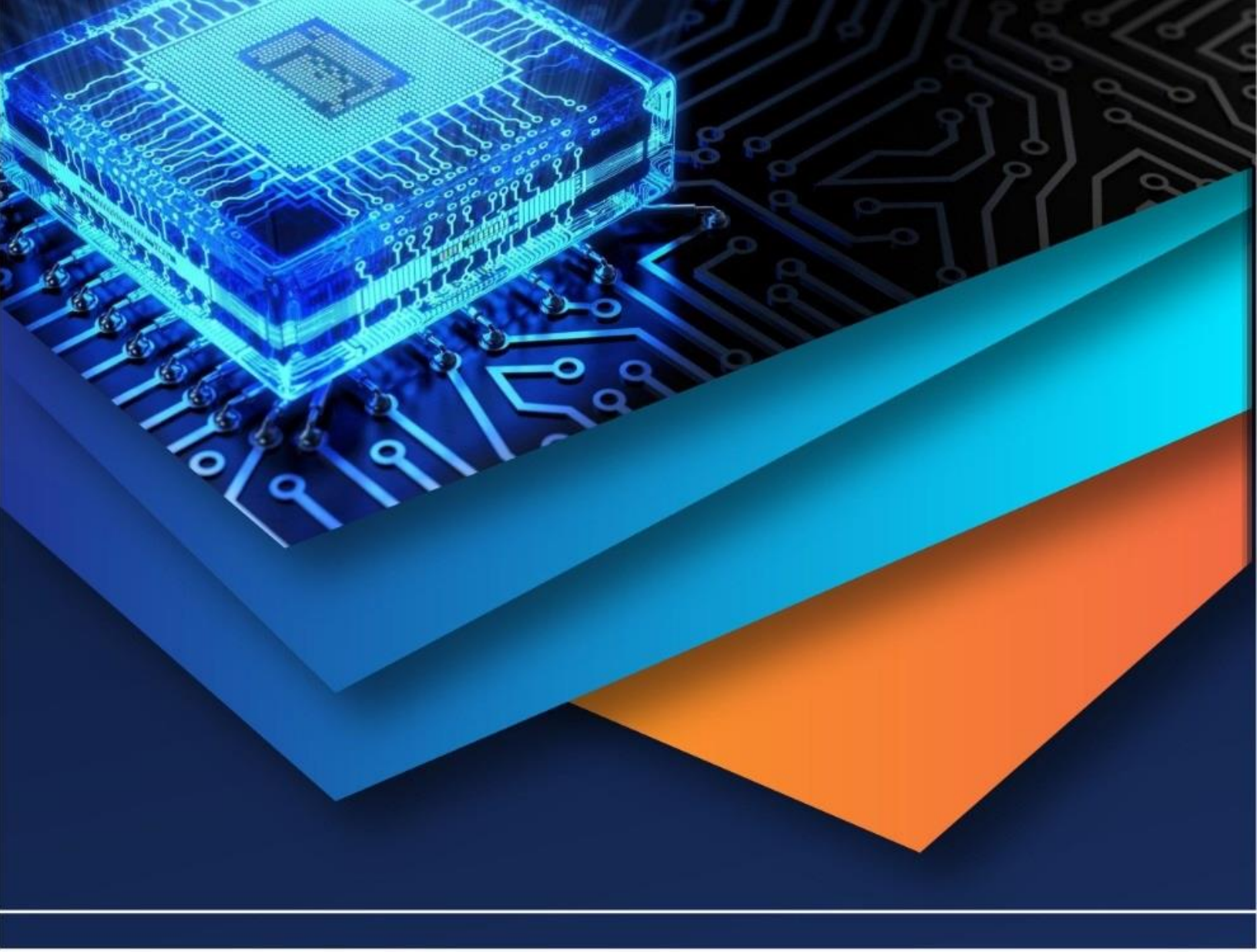
With 80% training data, all models show a measurable boost in performance, confirming improved generalization with larger training sets. CNN and LSTM gain consistent improvements, reaching accuracy values of 0.94 and 0.95 respectively. DNN benefits more significantly, rising to 0.96 across metrics due to richer feature learning. DBN remains the top performer, achieving approximately 0.97 in all metrics, demonstrating the advantage of pretraining-driven feature extraction. The clearer margin between models in this scenario highlights the increasing benefit of deep architectures when provided with more extensive training data.

V. CONCLUSION

This research confirms that DL architectures significantly enhance intrusion-detection performance on the NSL-KDD dataset compared to traditional machine-learning models. Among the four models, the DBN achieved the highest results, with accuracy, precision, recall, and F1-score of 0.96 in the 70–30 training–testing split and 0.97 in the 80–20 split. The DNN and LSTM models also showed strong performance (0.94–0.96), while the CNN provided a reliable baseline (0.92–0.94). The results indicate that increasing the training data proportion enhances feature representation and model generalization. Overall, the findings highlight the superiority of deep hierarchical, sequential, and nonlinear learning models for accurate, robust, and scalable intrusion detection in modern network environments.

REFERENCES

[1] Alashjaee, A.M. Deep learning for network security: an Attention-CNN-LSTM model for accurate intrusion detection. Sci Rep 15, 21856 (2025).
[2] Bhuiyan, M. D. S. M., Rafi, M. D. A. L., Rodrigues, G. N., Mir, M. N. H., Ishraq, A., Mridha, M. F., & Shin, J. (2025). Deep learning for algorithmic trading: A systematic review of predictive models and optimization strategies. Array, 26, 100390.
[3] Adefemi, K.O.; Mutanga, M.B. A Robust Hybrid CNN–LSTM Model for Predicting Student Academic Performance. Digital 2025, 5, 16.
[4] Singh, V., Sahana, S.K. &Bhattacharjee, V. A novel CNN-GRU-LSTM based deep learning model for accurate traffic prediction. Discov Computing 28, 38 (2025).
[5] Aman Kumar Singh, Gunjan Singh, Masood Husain Siddiqui, "NSE and BSE data set (2023-2025) ", IEEE Dataport, November 26, 2025.
[6] M. Mynuddin et al., "Automatic Network Intrusion Detection System Using Machine Learning and Deep Learning," 2024 IEEE International Conference on Artificial Intelligence and Mechatronics Systems (AIMS), Bandung, Indonesia, 2024, pp. 1-9.
[7] A. Modak and V. Dehalwar, "Design and Analysis of Intrusion Detection Using Deep Learning Models," 2024 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, 2024, pp. 1-6.
[8] N. K. Sah, M. Kolli, K. P. Dharmaraj and H. N. Vishwas, "Comparative Deep Learning Approach for Intrusion Detection," 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India, 2024, pp. 1-6.
[9] M. Moawad, V. W. Tze, P. T. Hang Hui and J. Y. Hsien Ming, "Deep Learning Based Intrusion Detection for Internet of Things and Edge Devices," 2023 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), Nadi, Fiji, 2023, pp. 01-06.
[10] A. Rathee, P. Malik and M. Kumar Parida, "Network Intrusion Detection System using Deep Learning Techniques," 2023 International Conference on Communication, Circuits, and Systems (IC3S), BHUBANESWAR, India, 2023, pp. 1-6.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY