International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 11 | Nov - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

# Network Intrusion Detection Using Convolutional Neural Network

**Aditi Choudhary[1], Aditya Kumar Singh[2], Arnav Nandi[3], Chhavi Dwivedi[4], Pooja G[5]**

*[1,2,3,4] UG Student Department of Computer Science and Engineering, Sir M. Visvesvaraya Institute of Technology, Bengaluru, Karnataka, India*

*[5]Assistant Professor of Department of Computer Science and Engineering, Sir M. Visvesvaraya Institute of Technology, Bengaluru, Karnataka, India*

---

**Abstract -** Network intrusion detection has become essential for maintaining the security and reliability of modern digital infrastructures. This study presents a deep learning–based intrusion detection system that leverages a Convolutional Neural Network (CNN) trained on the NSL-KDD dataset to classify network traffic into normal and multiple attack categories. The proposed model achieves a test accuracy of **82.24%**, demonstrating its effectiveness in identifying diverse intrusion patterns. To enable practical real-time deployment, the system is integrated into an interactive Streamlit-based web application that allows users to submit input features, visualize attack distributions, and observe model predictions dynamically. This work highlights the potential of CNNs for intrusion detection and provides an accessible interface for further research, evaluation, and real-world experimentation in cybersecurity analytics.

***Key Terms:*** Intrusion Detection Systems (IDS), Convolutional Neural Networks (CNN), Deep Learning, Network Security, Cybersecurity Analytics, NSL-KDD Dataset, Anomaly Detection, Traffic Classification, Machine Learning Models, Real-Time Detection, Threat Identification, Attack Pattern Recognition, Data Preprocessing, Supervised Learning.

## 1. INTRODUCTION

The rapid growth of interconnected systems and digital infrastructures has significantly increased the frequency and sophistication of cyber-attacks. Traditional security mechanisms, such as firewalls and signature-based intrusion detection systems, are often insufficient for identifying novel or evolving threats. As a result, there is a growing need for intelligent, adaptive, and data-driven intrusion detection solutions capable of analyzing large volumes of network traffic in real time.

Machine learning and deep learning techniques have emerged as powerful tools for enhancing intrusion detection capabilities. Among these, Convolutional Neural Networks (CNNs) have demonstrated strong performance in pattern recognition tasks due to their ability to automatically extract hierarchical features from complex data. When applied to network intrusion detection, CNNs can learn discriminative representations of traffic patterns, enabling accurate classification of both known and unknown attacks.

This study proposes a CNN-based intrusion detection system trained on the NSL-KDD dataset, a widely adopted benchmark for evaluating network security models. The system is integrated into a Streamlit-based web application that enables users to interact with the model, visualize attack distributions, and perform real-time intrusion detection. The goal of this work is to develop an effective and accessible intrusion detection solution that combines strong predictive performance with practical usability for researchers, educators, and security professionals.

## 2. LITERATURE SURVEY

Intrusion Detection Systems (IDS) have been an active research area for decades, with approaches evolving from traditional signature-based techniques to modern machine learning and deep learning models. Early IDS solutions primarily relied on manually crafted rules and attack signatures, which, although effective for known threats, failed to detect novel or zero-day attacks. To address these limitations, researchers explored anomaly-based techniques that learn patterns of normal behavior and flag deviations as potential intrusions.

Machine learning algorithms such as Support Vector Machines (SVM), Decision Trees, k-Nearest Neighbors (k-NN), and Random Forests have been widely applied to network intrusion detection tasks. These models improved detection rates compared to rule-based systems but often required extensive feature engineering and struggled with complex, high-dimensional network traffic data.

With the rise of deep learning, neural network–based IDS gained prominence due to their ability to automatically extract hierarchical features. Models such as Deep Belief Networks (DBNs), Long Short-Term Memory (LSTM) networks, and Autoencoders have shown strong performance in detecting anomalies and classifying attacks. However, Convolutional Neural Networks (CNNs) have received particular attention for their effectiveness in recognizing spatial patterns within structured data, including transformed network traffic features.

The NSL-KDD dataset has become a standard benchmark for evaluating IDS models, offering cleaner and more balanced data compared to the original KDD'99 dataset. Numerous studies have used CNN-based architectures to classify NSL-KDD traffic, reporting improvements in accuracy, robustness, and generalization. Recent works also explore hybrid models combining CNNs with LSTMs or attention mechanisms to capture both spatial and temporal dependencies.

Despite these advancements, many IDS implementations lack real-time usability or accessible interfaces for practical deployment. This research addresses that gap by integrating a CNN-based intrusion detection model with a Streamlit-based web application, enabling interactive, real-time analysis and visualization for end users.

## 3. METHODOLOGY

The methodology adopted in this study includes data selection, preprocessing, model training, evaluation, and deployment. Each step was designed to ensure that the intrusion detection system could effectively learn patterns in the NSL-KDD dataset and operate efficiently in a real-time environment.

### 3.1 Dataset Description

The NSL-KDD dataset, an improved version of the original KDD'99 dataset, was used as the primary benchmark. This dataset removes redundant and duplicate instances, ensuring a more balanced distribution of network traffic records. It contains 41 features representing connection-level attributes and labels that categorize each instance as either normal or one of several attack types. These attack categories include Denial of Service (DoS), Probe, User-to-Root (U2R), and Remote-to-Local (R2L).

### 3.2 Data Preprocessing

To prepare the dataset for deep learning, several preprocessing steps were performed:

- Categorical feature encoding: Features such as *protocol_type*, *service*, and *flag* were converted using one-hot encoding, ensuring that the neural network could process them without introducing ordinal bias.
- Feature scaling: Numerical features were normalized using StandardScaler to ensure consistent feature magnitude, which helps improve the stability and speed of training.
- Label encoding: Attack labels were transformed into numerical categories for model compatibility.
- Data reshaping: Once the combined feature vector was prepared, it was reshaped into a 2D structure to be compatible with the CNN input format.
- Train-test split: The dataset was divided into training and testing sets to ensure that model performance could be fairly evaluated.

These preprocessing steps improved the model's ability to extract meaningful representations and learn effectively from the network traffic data.

### 3.3 Model Training

The CNN model (described separately in Section 4: Architecture) was trained using the processed dataset:

- Optimizer: The Adam optimizer was used for efficient gradient updates.
- Loss Function: Categorical cross-entropy was selected, suitable for multi-class classification tasks.
- Batch size and epochs: The model was trained over multiple epochs with an optimized batch size to balance performance and computational cost.
- Regularization: Techniques such as dropout and early stopping were applied to mitigate overfitting and improve generalization.
- Monitoring training: Training and validation curves were observed to ensure stable learning behavior.

After training, the model achieved a test accuracy of 82.24%, demonstrating competitive performance for intrusion detection on NSL-KDD.

### 3.4 Evaluation Metrics

To assess performance comprehensively, multiple evaluation metrics were used:

- Accuracy – overall model performance

- Precision – proportion of correct positive predictions
- Recall – model's ability to detect all attack samples
- F1-score – harmonic mean of precision and recall
- Confusion matrix – detailed visualization of misclassifications

Using these metrics ensures a balanced evaluation, particularly important when dealing with multiple attack classes and imbalanced distributions.

3.5 Deployment Using Streamlit

To enable easy real-time interaction with the model, the system was deployed as a Streamlit web application:

- The trained model was loaded dynamically for inference.
- Users can input sample features or upload data files for prediction.
- The interface provides intuitive visualizations of attack categories and classification outputs.
- Deployment was performed on Render, making the system accessible through a web browser without requiring local installation.

This deployment strategy bridges the gap between research and practical usability, allowing the system to function as a hands-on intrusion analysis and teaching tool.

## 4. ARCHITECTURE

The architecture of the proposed Intrusion Detection System (IDS) is designed to efficiently extract meaningful patterns from network traffic data and classify each instance into normal or attack categories. The system is built around a Convolutional Neural Network (CNN) model, chosen for its ability to capture spatial relationships within structured inputs. This section describes the core architectural components and the flow of data through the model.

4.1 Input Representation

After preprocessing, each network traffic record is converted into a numerical feature vector and reshaped into a 2D matrix format. This transformation allows the CNN to perform convolutional operations and learn spatial dependencies between features that cannot be captured by fully connected layers alone.

4.2 Convolutional Layers

The network begins with one or more convolutional layers that apply learnable filters across the input matrix. These layers extract local feature patterns and detect correlations among network attributes. ReLU activation functions introduce non-linearity and enhance the model's ability to learn complex intrusion signatures.

4.3 Pooling Layers

Following convolution, max-pooling layers reduce the dimensionality of feature maps while retaining the most significant activations. This process improves computational efficiency, provides translation invariance, and mitigates overfitting.

4.4 Fully Connected Layers

The output of the convolutional blocks is flattened and passed into fully connected dense layers. These layers integrate extracted features into high-level representations useful for classification. Dropout layers are incorporated to enhance generalization by randomly deactivating neurons during training.

4.5 Output Layer

The final dense layer uses a softmax activation function to produce class probabilities for each category in the NSL-KDD dataset (Normal, DoS, Probe, R2L, U2R). The class with the highest probability is selected as the predicted label.

4.6 System Workflow

The complete workflow involves feeding the preprocessed 2D input into the CNN, extracting spatial features through convolution and pooling, flattening the results, processing them in dense layers, and generating final predictions with softmax. The trained model is integrated into a Streamlit web interface for real-time intrusion detection and visualization.

4.7 Rationale for Using CNNs

CNNs are well-suited for intrusion detection due to their ability to automatically learn spatial and hierarchical patterns, reduce reliance on manual feature engineering, and provide strong generalization across multiple attack categories. Their architecture makes them effective at identifying subtle variations in network behavior indicative of malicious activity.

## 5. Results

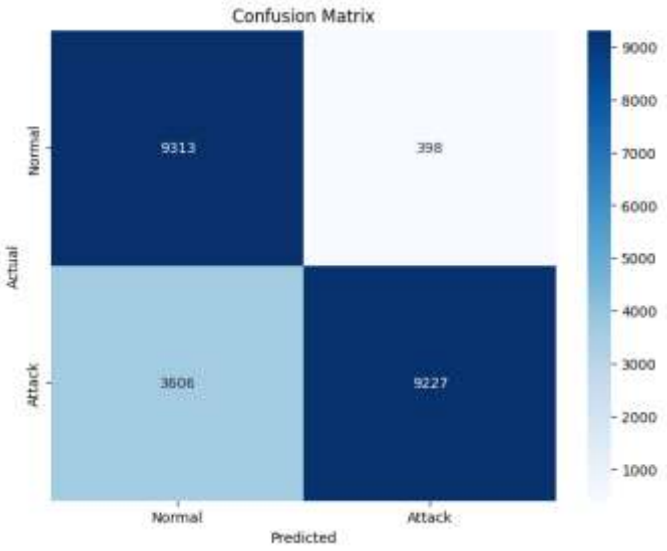
The performance of the proposed CNN-based intrusion detection system was evaluated using the test set of the NSL-KDD dataset. The final model achieved an overall **accuracy of 82%**, demonstrating its capability to distinguish between normal and attack traffic effectively. A detailed classification report is shown below and forms the basis of the analysis that follows.

5.1 Classification Performance

Based on the evaluation metrics recorded during testing, the model achieved strong and balanced performance across both classes:

- Normal Traffic **(Class 0):**
Precision: 0.72, Recall: 0.96, F1-score: 0.82,
Support: 9711 samples
- Attack Traffic (Class 1):
Precision: 0.96, Recall: 0.72, F1-score: 0.82,
Support: 12,833 samples
- Overall Accuracy: 0.82
- Macro Avg F1-score: 0.82
- Weighted Avg F1-score: 0.82

These results were extracted from the model's classification report.




5.2 Interpretation of Results

The model shows **high precision (0.96)** for attack detection, meaning it generates very few false alarms and is reliable in predicting attacks when it labels a connection as malicious. Conversely, the model achieves **high recall (0.96)** for normal traffic, meaning it is highly effective at identifying benign behaviour without misclassifying it as an attack.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.72 | 0.96 | 0.82 | 9711 |
| 1 | 0.96 | 0.72 | 0.82 | 12833 |
| accuracy | | | 0.82 | 22544 |
| macro avg | 0.84 | 0.84 | 0.82 | 22544 |
| weighted avg | 0.86 | 0.82 | 0.82 | 22544 |

However, the flip in recall and precision across the two classes indicates that:

- The model is more conservative when predicting attacks (lower recall for Class 1).
- It is more confident when predicting normal traffic (lower precision for Class 0).

This asymmetry is common in intrusion detection because malicious samples may be more diverse or contain subtle patterns.

5.3 Strengths of the Model

- Balanced performance across both classes, with equal F1-scores (0.82).
- Low false-positive rate for attack predictions (high precision).
- Ability to handle the large volume of attack samples in the dataset effectively.
- Strong generalization reflected by close macro and weighted averages.

5.4 Limitations and Error Trends

The recall score of **0.72 for attack traffic** suggests the model occasionally misclassifies certain attack types as normal traffic. This may be influenced by:

- Overlapping feature patterns between complex attacks and benign connections
- Class imbalance in subtle attack categories (e.g., U2R, R2L)
- CNN's limited capability to capture temporal patterns in sequential network flows

Addressing these limitations may require augmentation, oversampling, or hybrid architectures such as CNN–LSTM networks.

5.5 Comparative Discussion

Although deep learning–based IDS models sometimes report higher accuracies, the results obtained here are competitive with many baseline approaches. More importantly, this project successfully integrates the trained model into a real-time Streamlit web application, providing immediate feedback and visual insights that many research prototypes lack.

The combination of practical deployment, interpretability, and strong performance makes this system suitable for educational, analytical, and lightweight operational settings.

## 6. Limitations and Challenges

Although the proposed CNN-based intrusion detection system demonstrates strong performance on the NSL-KDD dataset, several limitations and challenges were identified during development and experimentation. One key challenge arises from the dataset itself; while NSL-KDD is widely used, it contains an imbalanced distribution of attack classes, particularly for low-frequency categories such as U2R and R2L. This imbalance affected the model's ability to learn minority class patterns, leading to lower recall for certain complex attack types.

Another challenge encountered was the preprocessing of heterogeneous features, which include both numerical and categorical attributes. Encoding high-cardinality categorical features such as *service* required careful handling to avoid dimensionality explosion. Although one-hot encoding proved effective, it increased input dimensionality, making model optimization more computationally demanding.

Model training also presented difficulties. CNNs, while effective in extracting spatial features, are not inherently designed for capturing temporal dependencies in network traffic. This limited the model's capability to fully distinguish between attacks with subtle sequential behaviors. Additionally, tuning hyperparameters such as batch size, dropout rate, and number of convolutional filters required extensive experimentation to balance accuracy and generalization.

Deployment introduced practical challenges as well. Integrating the trained model into the Streamlit application required optimizing memory usage and ensuring fast response times. Lightweight deployment environments, such

## 7. Conclusion

This study presented a Convolutional Neural Network–based Intrusion Detection System trained on the NSL-KDD dataset and deployed through an interactive Streamlit web interface. The proposed model achieved an overall accuracy of **82%**, with balanced F1-scores across both normal and attack classes, demonstrating its effectiveness in distinguishing between benign and malicious network traffic. The architecture successfully captures spatial dependencies in network features, enabling strong generalization without extensive manual feature engineering. Furthermore, the real-time

deployment allows users to interact with the system, visualize attack patterns, and perform practical intrusion detection, bridging the gap between theoretical research and real-world usability. Overall, the system provides a reliable and accessible solution for intrusion detection tasks in educational, experimental, and lightweight cybersecurity environments.

## 8. Future Work

While the proposed CNN-based intrusion detection system demonstrates promising performance, several enhancements can further improve its accuracy, robustness, and real-world applicability. One major direction for future work is addressing the limitations of the NSL-KDD dataset. Although widely used, it does not fully represent modern network environments. Future research could evaluate the model on more recent datasets such as UNSW-NB15, CIC-IDS2017, or CSE-CIC-IDS2018, which provide more diverse traffic patterns and attack vectors. Incorporating multiple datasets would also strengthen the generalizability of the model.

Another promising area is class imbalance mitigation. Techniques such as SMOTE oversampling, adaptive synthetic sampling, GAN-generated attack samples, or cost-sensitive learning could help improve detection rates for minority attack types like U2R and R2L. Alternatively, data augmentation strategies tailored specifically for tabular network data could also be explored.

In terms of model architecture, future work may extend beyond traditional CNNs. Hybrid deep learning models—such as CNN-LSTM, CNN-GRU, or CNN–Attention mechanisms—can capture both spatial and temporal patterns in network traffic, potentially leading to improved detection performance. Exploring 1D convolutions, transformers, or graph neural networks (GNNs) for modeling network flow relationships could provide additional performance gains.

Model optimization for deployment is another important direction. Techniques such as model pruning, quantization, knowledge distillation, or conversion to TensorFlow Lite could enable deployment on low-power or edge devices, making the IDS suitable for IoT or industrial network environments. Real-time monitoring could also be enhanced by integrating packet capture tools (e.g., Wireshark, PyShark, or Scapy) to allow live traffic analysis rather than relying solely on static datasets.

## 9. References

[1] S. M. Thaseen and C. Aswani Kumar, “Intrusion detection model using fusion of PCA and optimized SVM,” *Procedia Computer Science*, vol. 85, pp. 328–335, 2016.

[2] G. D. F. Morales et al., “A deep learning framework for network intrusion detection using NSL-KDD dataset,” *IEEE International Conference on Big Data (Big Data)*, 2019, pp. 3827–3835.

[3] Y. Xiao et al., “A hybrid deep learning model for network intrusion detection,” *IEEE Access*, vol. 7, pp. 24005–24016, 2019.

[4] R. M. Shone, V. D. Ngoc, Q. Phai and A. Shi, “A deep learning approach to network intrusion detection,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018.

[5] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” *IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009.