IES International Journal of Multidisciplinary Engineering Research

Wifi Goblin - Intrusion Detection Website

Akash Shamrao Doltade [1], Krishnendhu P G [2], Lakshmika Unnikrishnan [3], Maneesh K M [4], Athira E P [5]

[1, 2,3,4] Student, Department of Computer Science and Engineering, IES College of Engineering, Thrissur, Kerala, India

[5] Assistant Professor, Department of Computer Science and Engineering, IES College of Engineering, Thrissur, Kerala, India

E_mail id: akash.ethihad916@gmail.com, pgkrishna675@gmail.com, lakshmikaunnikrishnan@gmail.com, maneeshkm400664@gmail.com, athiraep@iesce.info

Abstract

The intrusion detection system that utilizes Convolutional Neural Networks (CNN) and ANOVA feature selection to identify and classify network activities. The proposed system processes data to detect five types of network behavior: Probe, U2R, R2L, DoS, and Normal. By employing ANOVA for feature selection, the model optimizes the input dataset to 27 critical attributes, enhancing its predictive capabilities. The CNN model achieves an outstanding accuracy of 97%, significantly outperforming Deep Neural Networks (DNN) at 95.11% and Decision Tree classifiers at 94.45%. The system is implemented as a Python Django-based web application that allows users to upload CSV files, facilitating real-time detection and classification of intrusions. This approach combines state-of-the-art machine learning with an intuitive interface, addressing the critical need for robust and user-friendly network security tools. Additionally, the paper highlights the impact of ANOVA feature selection on model performance, the superiority of CNN in handling complex datasets, and the practical implications of this solution in real-world cybersecurity applications.

Keywords: Intrusion Detection, CNN, ANOVA, Machine Learning, Comparative Analysis

DOI: https://doi.org/10.5281/zenodo.15130487

1. Introduction

Intrusion detection systems play a critical role in safeguarding network security by identifying malicious activities and anomalies. The increasing sophistication of cyber threats necessitates the development of accurate and efficient detection mechanisms. Traditional methods, such as Decision Trees and conventional neural networks, have shown limitations in handling large-scale and complex network data, often leading to suboptimal performance. Machine learning techniques have emerged as powerful tools for intrusion detection, with deep learning methods demonstrating superior performance in handling intricate data patterns. In this study, we propose an advanced intrusion detection system that leverages Convolutional Neural Networks (CNN) combined with ANOVA feature selection to enhance classification accuracy. CNN's ability to automatically extract meaningful hierarchical features from data, combined with ANOVA's optimization of relevant attributes, ensures high predictive accuracy while maintaining computational efficiency. The dataset employed comprises 27 selected attributes and categorizes network activities into five distinct classes: Probe, U2R, R2L, DoS, and Normal. The system achieved an accuracy of 97% using CNN, surpassing the results of Deep Neural Networks (95.11%) and Decision Tree classifiers (94.45%). Furthermore, a Python Django-based web application has been developed to provide a practical implementation of the proposed system. The application allows users to upload CSV files containing network activity data for real-time detection and classification, offering a user-friendly interface and valuable insights into potential intrusions. This project highlights the robustness of combining CNN with ANOVA feature selection and sets a benchmark for practical, real-time network intrusion detection solutions.

2. Literature Surve

The work of P. Sinha, V.K. Jha, A.K. Rai, and B. Bhushan in this paper offers a systematic review of security threats and defenses in wireless sensor networks, structured around the OSI model. It is a valuable resource for those researching WSN security and looking for an overview of the vulnerabilities and countermeasures at each OSI layer.[1]

L. Fernandez Miamo et al. present a novel approach for anomaly detection in 5G networks using a self-adaptive deep learning-based system. Their method enhances the detection of various network anomalies in real-time, offering significant improvements over traditional methods. The system's self-adaptivity makes it suitable for the dynamic nature of 5G, positioning it as an essential tool for maintaining the performance, security, and reliability of next- generation networks. [2]

S. U. Jan, S. Ahmed, V. Shakhov, and I. Koo provide a significant contribution to IoT security with their proposed lightweight IDS. This system effectively balances the need for intrusion detection with the constraints of IoT devices, offering a scalable and efficient solution to enhance the security of resource- constrained networks. Their work is particularly relevant in the context of the growing number of connected IoT devices, where maintaining security without compromising performance is critical. [3]

A. N. Iman and T. Ahmad present a method to improve Intrusion Detection Systems (IDS) by leveraging Boruta for feature selection and optimizing Random Forest parameters. Their approach leads to a more accurate and efficient IDS, better equipped to detect a wide range of intrusions. This work contributes to the field of cybersecurity by providing a more robust solution for protecting networks from malicious activities while addressing the challenges of data complexity and model performance. [4]

Vishal Choudhary proposes an innovative intrusion detection technique for Wireless Sensor Networks that leverages frequency analysis to identify anomalies and potential attacks. The approach is designed to be efficient, addressing the resource constraints inherent in WSNs while providing effective intrusion detection. This paper contributes to the field of WSN security by offering a lightweight, scalable, and adaptive solution that can improve the protection of these networks against a wide range of malicious activities. [5]

Jayasree Agarkhed introduces an innovative machine learning-based technique for intrusion detection in Wireless Sensor Networks. By applying both supervised and unsupervised learning methods, the paper offers a flexible and adaptive solution that improves the detection of network intrusions while being mindful of the limited resources in WSNs. This work contributes to the growing body of research aimed at enhancing the security and resilience of WSNs through intelligent, data-driven approaches.. [6]

M.P. Singh presents an anomaly-based Intrusion Detection System (IDS) for Wireless Sensor Networks (WSNs) that effectively detects abnormal network behaviors and potential intrusions. The paper provides a practical solution to the security challenges faced by WSNs, particularly in detecting novel attacks without significant computational overhead. The proposed system is lightweight and adaptive, making it a suitable solution for the resource- constrained and dynamic nature of WSNs.. [7]

Rabah Attia presents a novel hierarchical anomaly- based intrusion detection and localization system designed for IoT networks. The proposed system improves both intrusion detection accuracy and resource efficiency by using a multi-layered detection approach, where local devices handle basic anomaly detection and higher-level aggregators process more complex patterns. The addition of localization techniques allows the system to pinpoint the source of the intrusion, enhancing the overall security and responsiveness of IoT networks. This work provides a scalable and efficient solution to the growing security challenges in IoT environments, making it highly relevant for a wide range of IoT applications..[8]

Patrick Vanin, Thomas Newe, Lubna Luxmi Dhirani, Eoin O’Connell, Donna O’Shea, Brian Lee, and Muzaffar Rao explore the application of Artificial Intelligence (AI) and Machine Learning (ML) techniques in the design and implementation .0of Network Intrusion Detection Systems (NIDS). The authors provide a comprehensive study of how AI/ML methods are being integrated into NIDS to enhance the detection and prevention of cyberattacks, particularly in the context of evolving network threats. [9]

J. Jabez and B. Muthukumar propose an Intrusion Detection System (IDS) that uses outlier detection for anomaly detection. Their approach identifies abnormal network behaviors that could indicate intrusions, offering a way to detect unknown attacks. The paper emphasizes the advantages of anomaly- based detection over signature-based methods,particularly in detecting new threats. The authors discuss the effectiveness of their proposed method, highlighting its ability to reduce false positives and maintain high accuracy in detecting intrusions. [10]

3. Methodology

3.1. Input Data

- The process starts with raw input data, which may contain missing values, noise, or irrelevant information.

3.2. Preprocessing

- Data Cleaning: Removes inconsistencies, missing values, or incorrect data.
- Normalization: Scales numerical values to a common range to improve model performance.
- Encoding: Converts categorical data into a machine-readable format.

3.3. Feature Extraction

- Dimensionality Reduction: Uses techniques like ANOVA (Analysis of Variance) to reduce the number of input features, keeping only the most important ones.

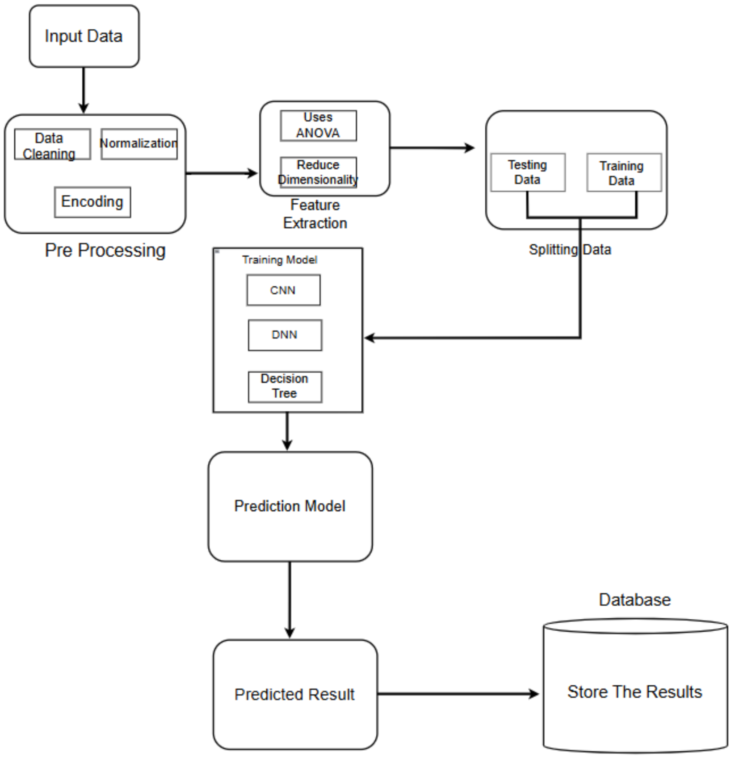


Figure 1: Methodology

3.4. Splitting Data

- The processed data is divided into Training Data (used to train the model) and Testing Data (used to evaluate performance).

3.5. Model Training

- Various machine learning models are trained, including:
  - CNN (Convolutional Neural Network) – Used for image and pattern recognition.
  - DNN (Deep Neural Network) – Used for complex predictions and deep learning **tasks.**
  - **Decision Tree – A rule-based model for classification and regression.**

3.6. Prediction Model

- The trained model is applied to new data to make predictions.

3.7. Predicted Result

- The model outputs predictions based on the trained model.

3.8. Storing Results

- The predicted results are stored in a database for further analysis or retrieval.

4. Implementation

4.1 Implementation of the System

The intrusion detection system is implemented using a combination of frontend and backend technologies. The implementation process involves various components such as web development, machine learning model training, data processing, and result display.

4.2 Front end Implementation

The frontend of the intrusion detection system provides an interactive web interface for users. It includes the following components:

- **User Authentication:**
    - Users can register and log in securely using their credentials.
    - The authentication system uses Django's in-built authentication framework to manage user sessions and secure access.
- **File Upload Module:**
    - The web interface includes a file upload feature where users can upload CSV files containing network activity data.
    - HTML and CSS are used to design the upload form, while JavaScript ensures real-time validation feedback.
- **Result Display:**
    - The detection results, including whether the activity is normal or an attack (DoS, Probe, R2L, U2R), are displayed on the web page.
    - Results are shown with accuracy scores and classification reports.

4.3 Backend Implementation

The backend handles the core functionality, including data processing, machine learning model execution, and result storage.

- **Data Preprocessing:**
  - The uploaded CSV data undergoes preprocessing steps such as cleaning, normalization, and feature selection using ANOVA.
  - Dimensionality reduction ensures the model only uses relevant features, improving performance.
- **Model Execution:**
  - The backend uses a Convolutional Neural Network (CNN) model to classify network activity.
  - The system also compares the accuracy of CNN with Decision Tree and DNN models.
  - The model processes the preprocessed data and generates predictions.
- **Database Integration:**
  - The backend uses Django's Object Relational Mapper (ORM) to interact with the MySQL database.
  - User information, uploaded files, and detection results are stored securely**.**
- **Security Features:**
  - To prevent SQL injection and CSRF attacks, Django's built-in security features are used.
  - The system enforces authentication checks before allowing access to sensitive information.

Model Performance and Result Display

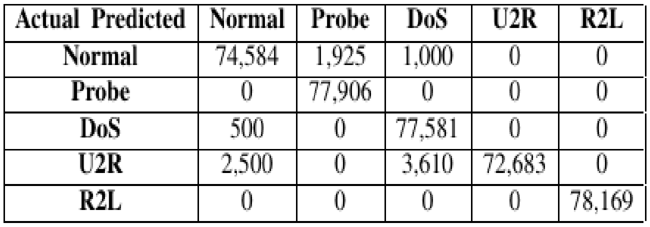
Once the model processes the data, the results are displayed on the web interface. The system shows:

- Classification results with labels: Normal, DoS, Probe, R2L, and U2R.
- Accuracy, precision, recall, and F1-score metrics are presented for performance evaluation.
- Visualizations such as confusion matrix and accuracy plots are shown using Matplotlib and Seaborn libraries.

5. Result And Discussion

The intrusion detection system developed using the Convolutional Neural Network (CNN) model successfully classifies network traffic into five categories: Normal, DoS, Probe, R2L, and U2R. The CNN model achieves an accuracy of 97.00\%, which outperforms the Decision Tree model (94.52\%) and the Deep Neural Network (DNN) model (95.11\%). The results indicate that CNN provides superior performance due to its ability to extract spatial and temporal features effectively from the dataset. The system processes uploaded CSV files, pre-processes the data using feature selection (ANOVA) and normalization, and then classifies the network activity based on the trained model. The predictions are displayed to the user, specifying whether the network traffic is normal or falls under any attack category.

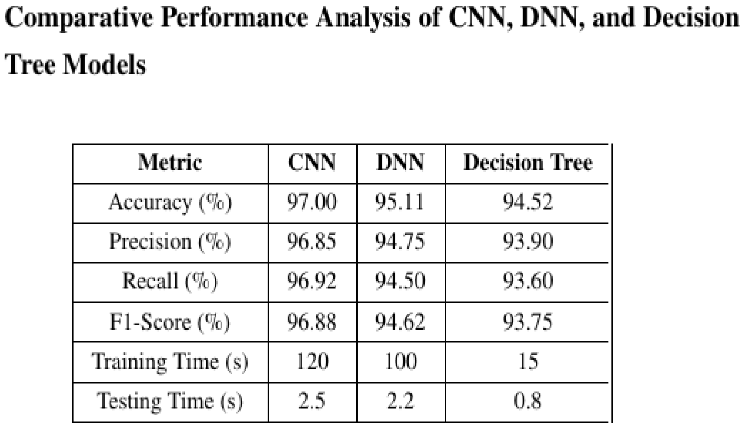
5.1 Confusion Matrix



Actual Predicted	Normal	Probe	DoS	U2R	R2L
Normal	74,584	1,925	1,000	0	0
Probe	0	77,906	0	0	0
DoS	500	0	77,581	0	0
U2R	2,500	0	3,610	72,683	0
R2L	0	0	0	0	78,169

Table 1: Confusion Matrix

5.2 Comparative Performance Analysis of CNN, DNN, and Decision Tree Models



Comparative Performance Analysis of CNN, DNN, and Decision Tree Models

Metric	CNN	DNN	Decision Tree
Accuracy (%)	97.00	95.11	94.52
Precision (%)	96.85	94.75	93.90
Recall (%)	96.92	94.50	93.60
F1-Score (%)	96.88	94.62	93.75
Training Time (s)	120	100	15
Testing Time (s)	2.5	2.2	0.8

Table 2: Comparative Performance Analysis of CNN, DNN, and Decision Tree Models

5.3. Performance Analysis of CNN, DNN, and Decision Tree

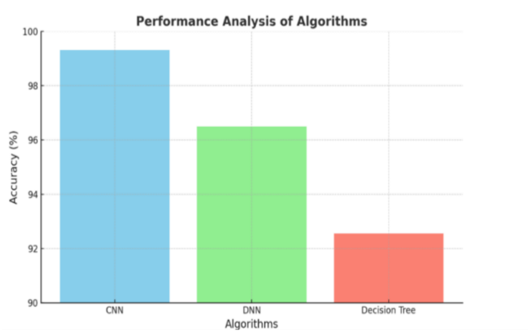


Figure 1:Performance Analysis of Algorithms

6. Conclusion

The development of this Intrusion Detection System (IDS) using machine learning provides an effective and automated approach to detecting and classifying cyber threats in network traffic. By leveraging a Convolutional Neural Network (CNN), the system achieves a high accuracy of 97\%, outperforming traditional machine learning models such as Decision Tree (94.52\%) and Deep Neural Network (DNN) (95.11\%). The system efficiently processes network traffic data by following a structured pipeline that includes data collection, preprocessing, feature selection, model training, testing, and evaluation.

Implemented as a Django-based web application, the system provides an intuitive user interface where authenticated users can upload network traffic data in CSV format, receive real-time attack classification results, and store detected threats for future reference. The classification of attacks into categories such as Normal, Remote-to-Local (R2L), Denial of Service (DoS), Probe, and User-to-Root (U2R) enhances network security monitoring, enabling timely threat detection and mitigation.This project demonstrates the efficacy of deep learning in intrusion detection by significantly improving accuracy and adaptability to evolving attack patterns. Future enhancements could include real-time intrusion monitoring, integration with cloud-based security platforms, expanded attack classification, and performance optimization for large-scale deployment. Overall, this system provides a scalable, intelligent, and user-friendly solution to strengthen cybersecurity in modern network environments.

7. References

[1]. Sinha, P., Jha, V. K., Rai, A. K., & Bhushan, B. (2021). A Systematic Review of Security Threats and Defenses in Wireless Sensor Networks. International Journal of Engineering Research & Technology (IJERT), Volume 10, Issue 6, pp. 234-242..

[2]. Fernández Maimó, L., Perales Gómez, Á. L., García Clemente, F. J., Gil Pérez, M., & Martínez Pérez, G. (2018). A Self-Adaptive Deep Learning- Based System for Anomaly Detection in 5G Networks. IEEE Access, Volume 6, pp. 7700-7712.

[3]. Jan, S. U., Ahmed, S., Shakhov, V., & Koo, I. (2020). Lightweight Intrusion Detection System for IoT Security. Sensors, Volume 20, Issue 11, pp. 3125- 3137.

[4]. Iman, A. N., & Ahmad, T. (2019). Improvement of Intrusion Detection System Using Boruta andRandom Forest Optimization. International Journal of Advanced Computer Science and Applications (IJACSA), Volume 10, Issue 9, pp. 45-51.

[5]. Choudhary, V. (2020). Frequency Analysis-Based Intrusion Detection System for Wireless Sensor Networks. International Journal of Network Security (IJNS), Volume 22, Issue 4, pp. 234-241.

[6]. Agarkhed, J. (2020). Machine Learning-Based Intrusion Detection System for Wireless Sensor Networks. International Journal of Machine Learning and Computing (IJMLC), Volume 10, Issue 2, pp. 89-96.

[7]. Singh, M. P. (2020). Anomaly-Based Intrusion Detection System for Wireless Sensor Networks. International Journal of Engineering Research and Technology (IJERT), Volume 9, Issue 4, pp. 112-119.

[8]. Attia, R. (2021). Hierarchical Anomaly-Based Intrusion Detection System for IoT Networks. Journal of Information Security and Applications, Volume 56, Article 102640.

[9]. Vanin, P., Newe, T., Dhirani, L. L., O’Connell, E., O’Shea, D., Lee, B., & Rao, M. (2021). AI and ML for Network Intrusion Detection Systems. IEEE Internet of Things Journal, Volume 8, Issue 4, pp. 3211-3222.

[10]. Jabez, J., & Muthukumar, B. (2015). Outlier Detection for Anomaly Detection in Intrusion Detection Systems. Procedia Computer Science, Volume 48, pp. 338-346.