Network Based Intrusion Detection using Convolutional Neural Network

Dr Brindha S[1], Ms. Dhamayanthi A[2], Mr. Amish Peniel A[3], Mr. Harish Raj V[4]

Mr. Hassan A[5], Mr. Praveen M[6], Mr. Pranith A[7]

[1]Head of the Department, Computer Networking, PSG Polytechnic College, Coimbatore
[2]Lecturer, Computer Networking, PSG Polytechnic College, Coimbatore
[3,4,5,6,7] Students, Computer Networking, PSG Polytechnic College, Coimbatore

---***---

Abstract - The diversification of wireless network traffic attack characteristics has led to the problems what traditional intrusion detection technology with high false positive rate, low detection efficiency, and poor generalization ability. In order to enhance the security and improve the detection ability of malicious intrusion behaviour in a wireless network, this paper proposes a wireless network intrusion detection method based on convolutional neural network (CNN). First, the network traffic data is characterized and pre-processed, then modelled the network intrusion traffic data by CNN. The low-level intrusion traffic data is abstractly represented as advanced features by CNN, which extracted autonomously the sample features, and optimizing network parameters to converge the model. Finally, we conducted a sample test to detect the intrusion behaviour of the network. during the training process, and the key feature information loss and parameter tuning difficulty are easily caused during the training process. This paper considers using the end-to end semi-supervised network training classifier of convolutional neural network (CNN), and the multi-layer feature of CNN to detect network, learn the sample features and discover the rules in the data training process to simplify the implementation process.

Key Words: intrusion detection, anomaly detection, deep learning, convolution neural network, network security.

1. INTRODUCTION

In today's digital era, networks have become an integral part of our personal and professional lives. They enable seamless communication and data transfer across the globe. However, this connectivity also introduces numerous security threats that can compromise sensitive information, disrupt services, or cause financial loss.

To mitigate these risks, **cybersecurity** techniques are employed. Cybersecurity refers to a range of strategies, tools, and practices designed to protect networks, devices, and data from unauthorized access, attacks, and damage. Common cybersecurity measures include **antivirus software**, **firewalls**, and **encryption protocols**. While these methods provide a strong defence, they may struggle to identify and respond to newer, more sophisticated attacks that exploit emerging vulnerabilities.

To strengthen network security, **Intrusion Detection Systems (IDS)** have been introduced. An IDS plays a crucial role in identifying, monitoring, and analysing suspicious activities within a network. It detects potential security breaches, unauthorized access attempts, and other malicious behaviour by analysing traffic patterns and system activities. IDS solutions are designed to provide real-time alerts and detailed reports, enabling administrators to respond swiftly to threats.

Figure 1 illustrates an overview of the intrusion detection process. In this example, a **firewall** functions as an intrusion detector. Positioned at the network's entry point, the firewall filters incoming and outgoing traffic. It inspects data packets based on predefined security rules and blocks potentially harmful traffic before it reaches the internal network. This proactive filtering mechanism helps prevent attacks such as malware infections, denial-of-service (DoS) attacks, and unauthorized access attempts.

While firewalls and IDS solutions are key components of network security, they are most effective when combined with other security strategies such as **intrusion prevention systems (IPS)**, **multi-factor authentication (MFA)**, and **regular security audits**. Together, these measures create a comprehensive defence framework that minimizes vulnerabilities and ensures robust protection against evolving cyber threats.

2. Related Work

This chapter gives a survey of literature work done by other researchers. I've learned some existing techniques from their research work, few of them are discussed below.

Gharaee and Hossein [2] proposed a genetic algorithm and SVM with a new feature selection technique to improve the IDS. The new feature selection method based on a genetic algorithm with innovative fitness function to increase the true positive rate and reduce the false positive simultaneously reduces the time taken for execution.

Gul and Adali [3] proposed a feature selection process for Intrusion Detection. Feature selection is an important process before classification is performed. When selecting the important feature it will reduce the execution time and increase the accuracy of the model.

Zhang and Wang [4] proposed an effective wrapper based feature selection to in- crease the accuracy of the algorithm. The wrapper method feature selection is based on Bayesian Network classifier.

Moustafa et al, [5] compared the signature based network intrusion detection that Anomaly based detection is more efficient. Anomaly does not follow patterns like signature based detection. The Authors evaluate their classification algorithm with two bench- mark datasets of Network Intrusion Detection System (NIDS) NSL-KDD and KDD99 and find out that the datasets may be lacking in accuracy because of poor recent attack types, so the author used UNSW NB15 dataset. The author shows that evaluation of UNSW NB15 is done in three aspects to find its complexity. Also the system designed by [6] offered higher accuracy based on optimization in real time.

Intruders use more enhanced techniques to break the security so enhancement in IDS is needed . Primartha and Tama [7] used three different (UNSW NB15, GPRS, and NSL-KDD) datasets to perform classification process using Random forest, Naive Bayes, and Neural Network to get high accuracy and low warning rate and K-cross validation is done.

3. Proposed Algorithm

In existing machine learning based IDS, always depending on the previous data may not be effective for newly generated attacks. The proposed deep learning model is dynamic and it can also be used for unusual patterns.

3.1. Convolution Neural Network (CNN)

In this proposed work Convolution Neural Network (CNN) used as a learning model for classification in IDS. Convolution Neural Networks (CNN) is the kernel convolves across all the elements on the input tensor to construct the elements of the feature map for that kernel. An arbitrary number of feature maps can be obtained by implementing the convolution operation with different kernels. While training, the convolution operation is called forward propagation; during back propagation, the gradient descent optimization technique updates the learnable parameters (kernels and weights) according to the loss value. A pooling layer provides a typical down sampling operation to reduce the dimensionality of the feature maps to introduce translation invariance to small shifts and distortions and thereby decrease the number of subsequent learnable parameters. The pooling function is pool(·); for each feature j is a local neighborhood around location (i, j). The fully connected layers are the final outputs of the CNN, such as the probabilities for each class in classification tasks. The number of output nodes in the final fully connected layer is usually equal to the number of classes. A nonlinear function, such as ReLU, follows each fully connected layer. Finally, a loss function is calculated to assess the compatibility of the CNN's forward propagation output predictions with the provided ground truth labels.

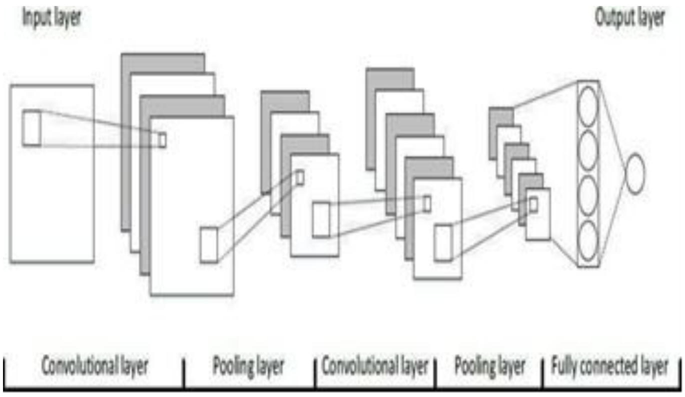

Figure 1. CNN Layers

The loss function for CNN optimization is given by:

$$L = \frac{1}{N}\sum\nolimits_{n=1}^{N}(y_n \log(y_n) + (1 - y_n)\log(1 - y_n\ ))$$

Figure 2. Loss function for CNN optimization

where N is the number of samples, yn is the actual label, and yn is the predicted probability
Training a CNN determines the global minima, which identify the best-fitting set of parameters by minimizing the loss function.

3.2. System Architecture

Figure 3 represents the system architecture. when the dataset is given as an input and the given input splitted into trained and test data. CNN algorithm applied to the splitted data and we get a trained model. That trained model and test data are compared and we performed evaluation metrics for that data.

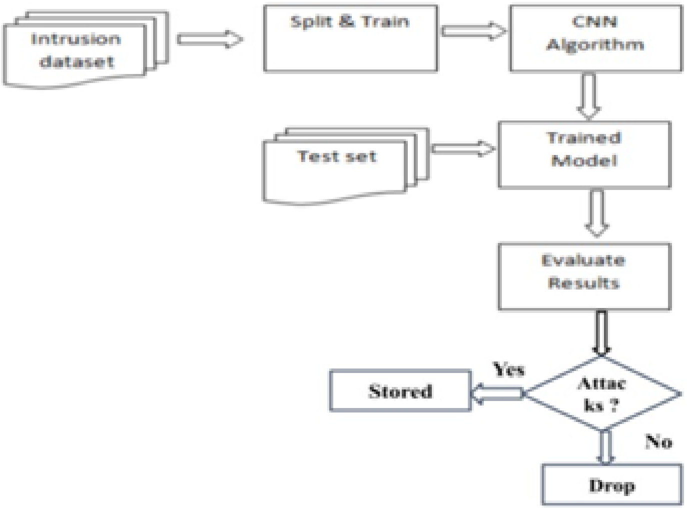

Figure 3. Architecture Diagram

4. Functional Requirement

Here the dataset is taken from kdd. The downloaded dataset contains train and test data and the outputs are classified into different classes with binary value “0” and ”1” for normal and attacked data. The train dataset is considered as the train set and the test dataset is considered as the test set. CNN is applied for the classification process in this work and evaluates the algorithm with performance metrics.

4.2. Dataset Detail

The dataset contains 1,75,341 recorded datas as training data and 82,332 recorded datas as testing data. It has 49 features and it is categorized into six groups like flow, time, content, etc. It recorded 9 different types of recent and common attacks like Dos, fuzzers, backdoors, worms, etc. the output categorized into binary values as “0” and “1” for attack and normal data.

4.2.1 Data Collection

Data collection is a crucial process that involves gathering relevant data to support the implementation and training of machine learning models. The purpose of this phase is to acquire high-quality data that can contribute to the accuracy and reliability of the system. This involves identifying various sources of data, including open-source datasets, proprietary datasets, and real-time data streams. The collected data must be examined for completeness, consistency, and relevance to ensure it aligns with the objectives of the intrusion detection system. Additionally, data collection may involve techniques such as web scraping, sensor data acquisition, or retrieving logs from network traffic monitoring tools. Once the data is collected, it undergoes an initial assessment to determine its suitability for further processing and analysis.

4.2.2 Data Visualization

Data visualization is an essential component of data analysis that enhances interpretability and comprehension of complex datasets. The primary purpose of visualization is to present the data in a graphical format that allows stakeholders to identify patterns, trends, and insights effectively. Various visualization techniques such as bar graphs, line charts, scatter plots, and heatmaps can be employed to display network intrusion detection results. For this study, we use graphical representation to illustrate the accuracy of the learning model utilized in intrusion detection. This enables users to visually analyze model performance, detect anomalies, and interpret key findings efficiently. Data visualization also aids in debugging and refining models by providing an intuitive representation of performance metrics and feature distributions.

4.2.3 Data Preprocessing

Data preprocessing is a critical step in preparing raw data for machine learning algorithms. Raw data often contains inconsistencies, missing values, and irrelevant features, which can impact the performance of the model. To address these challenges, data preprocessing involves the following steps:

- **Normalization:** This step ensures that all features are scaled within a specific range, preventing large numerical values from dominating smaller ones. Normalization techniques like Min-Max Scaling and Z-score normalization are applied to standardize data distributions.
- **Handling Missing Values:** Incomplete data is processed using imputation techniques, such as filling missing values with the mean, median, or mode.
- **Encoding Categorical Features:** Since machine learning models operate on numerical data, categorical variables (e.g., protocol types, attack labels) are transformed into numerical representations using encoding techniques like One-Hot Encoding or Label Encoding.
- **Noise Removal:** Redundant or irrelevant data points that may interfere with learning are filtered out to enhance model accuracy.

4.2.4 Dataset Splitting

After preprocessing, the dataset is divided into separate subsets to train and evaluate the model effectively. The dataset is split into the following parts:

- Training Data: This subset is used to train the model and help it learn patterns and features from the data.
- Testing Data: A portion of the dataset is set aside for evaluating the trained model’s performance on unseen data.
- Validation Data (Optional): Sometimes, an additional validation set is created to fine-tune the model parameters before final testing.

4.2.5 Model Training

Model training is the process of applying a deep learning model to the prepared training dataset to learn from it. During training:

- The dataset is fed into the model, which processes input features through multiple layers of neurons (in the case of deep learning models).
- The model iteratively adjusts its internal parameters using optimization algorithms such as Stochastic Gradient Descent (SGD) or Adam Optimizer.
- Loss functions, such as Mean Squared Error (MSE) or Cross-Entropy Loss, measure the difference between predicted and actual values.
- The model refines itself through multiple epochs, learning patterns and relationships within the dataset.
- Techniques like dropout regularization and batch normalization are applied to prevent overfitting and improve model generalization.

4.2.6 Model Evaluation

Once the model is trained, it is evaluated using the testing dataset to measure its performance. The evaluation process involves:

- **Comparing Predictions with Actual Values:** The model's output is compared with the ground truth to determine accuracy and effectiveness.
- **Performance Metrics Calculation:** Various statistical metrics are used to evaluate model performance:
  - **Mean Squared Error (MSE):** Measures the average squared difference between actual and predicted values.
  - **Mean Absolute Error (MAE):** Computes the average absolute difference between predictions and actual results.
  - **R-Square (R^2):** Determines how well the model explains the variance in the dataset.

- **Root Mean Squared Error (RMSE):** Provides a standardized measure of prediction error.

Based on these metrics, necessary improvements and fine-tuning can be implemented to enhance the model's performance further. The evaluation phase ensures that the deep learning model is reliable and robust before deployment.

5. Result

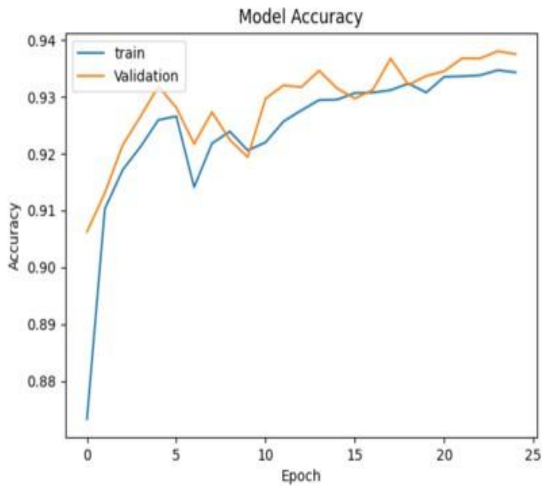

Figure 4. CNN Model Accuracy

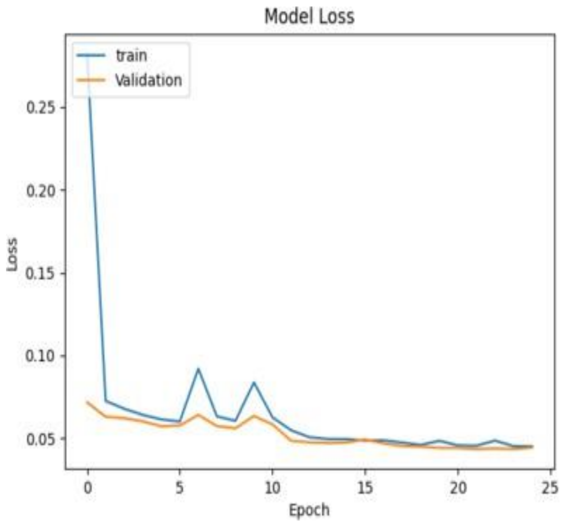

Figure 5. CNN Model Loss

The proposed work is done with a deep learning model to improve the Intrusion Detection System. Here, CNN deep learning model is used to find the accuracy of IDS from UNSW NB15 dataset. The proposed work is done in python 3.7 with libraries of keras, tensorflow, matplotlib and other mandatory files. Here we get 93.5% of algorithm accuracy using CNN and Evaluation metrics calculated to find the performance of the algorithm. The result shows that intrusion detection is efficient using CNN algorithm. The Figure 4 shows the model accuracy of training and validation datas by using CNN algorithm with 25 epoch.

The Figure 5 shows the model loss accuracy of training and validation set using CNN algorithm with 25 epoch. The Figure 6 shows the evaluation metrics of CNN algorithm to analyze the performance of the model.

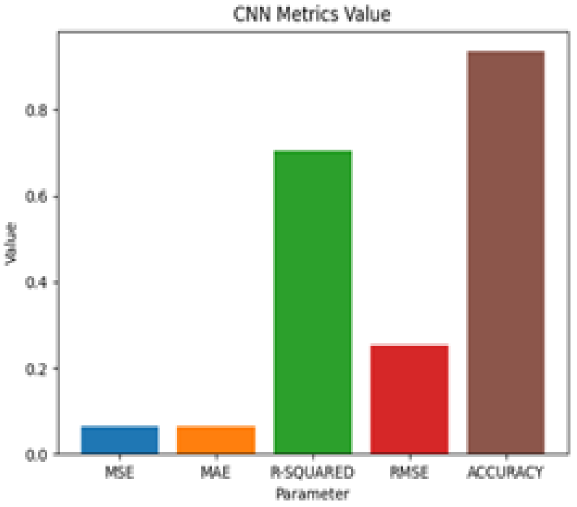

Figure 6. CNN Evaluation Metrics.

Evaluation metrics are used to survey the classification of the statistical learning model. Evaluating the learning models or algorithms is consequential for any project. There are many distinct types of evaluation metrics available to test a model. Mean Square Error (MSE) is a mean of Squared Error it is the difference between actual and predictive value. Mean Absolute Error (MAE) measures the difference between two variables and absolute error of each prediction error. R-Squared measures the goodness of fit of a regression model. Root Mean Square Error (RMSE) measures the square root of MSE value.

5. Conclusion

The proposed work is to improve intrusion detection efficiency though we have many existing IDS mostly developed in the Machine learning algorithm that fails to provide strong IDS to prevent from newly formed attacks because it mostly depends on previous data. Here, CNN deep learning model is used for developing the IDS. By using UNSW NB15 network intrusion public Dataset we perform the classification technique by applying CNN algorithm and we get 93.5% accuracy. The accuracy shows that CNN is efficient in Intrusion detection and evaluation metrics also performed to analyze the performance of the model.

6. Future Works

In the future, this research can be further enhanced by improving feature selection, developing hybrid deep learning models, and optimizing real-time deployment in network security environments. The robustness of CNN-based intrusion detection systems can be strengthened by addressing adversarial attack resistance and ensuring cross-dataset generalization. Additionally, incorporating explainability techniques and extending the model to emerging areas like cloud and IoT security can enhance its practical applicability. The following key areas highlight potential directions for future improvements:

Enhanced Feature Engineering

- Incorporate advanced feature selection techniques such as Autoencoders or Genetic Algorithms to optimize the input data representation.
- Explore temporal features to better capture sequential attack patterns in network traffic.

Hybrid Deep Learning Models

- Combine CNN with other deep learning models like LSTMs, GRUs, or Transformers to improve anomaly detection, especially for sophisticated attacks.
- Develop ensemble models by integrating CNN with traditional machine learning algorithms (e.g., Random Forest, SVM) for enhanced accuracy.

Real-Time Detection & Deployment

- Optimize the trained model for real-time intrusion detection in high-speed networks using lightweight architectures.
- Deploy the model in a live network monitoring environment using IDS tools like Snort or Suricata.

Adversarial Attack Resistance

Investigate the robustness of CNN-based NIDS against adversarial attacks and develop countermeasures such as adversarial training or anomaly detection techniques.

Cross-Dataset Generalization

○ Evaluate the model's performance on multiple benchmark datasets (e.g., NSL-KDD, CICIDS2017, UNSW-NB15) to enhance its adaptability and robustness across different network environments.

Explainability and Interpretability

○ Integrate explainable AI (XAI) techniques to provide human-readable insights into model decisions, improving trust and usability in cybersecurity applications.

Cloud & IoT Security Applications

○ Extend the model to detect intrusions in cloud-based networks and Internet of Things (IoT) environments, addressing challenges like resource constraints and scalability.

8. Acknowledgment

We extend our deepest gratitude to Ms. A. Dhamayanthi for their invaluable guidance, encouragement, and constructive feedback throughout this research. Their expertise and insights have been instrumental in shaping the direction and quality of this work.

We also acknowledge the support provided by PSG Polytechnic College, whose resources and infrastructure significantly contributed to the successful completion of this study. Special thanks to our colleagues and peers for their valuable discussions, suggestions, and motivation throughout the research process.

Furthermore, we express our appreciation to the authors and contributors of publicly available datasets and research literature, which served as a foundation for our work. Lastly, we are grateful to our families and friends for their unwavering encouragement and support during this journey.

References

[1] Coelho F, de Pa´dua Braga A, Verleysen M. Cluster homogeneity as a semi-supervised principle for feature selection using mutual information. In European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning 2012 (pp. 507-512).

[2] Gharaee H, Hosseinvand H. A new feature selection IDS based on genetic algorithm and SVM. In2016 8th International Symposium on Telecommunications (IST) 2016 Sep 27 (pp. 139-144). IEEE.

[3] Gu¨l A, Adalı E. A feature selection algorithm for IDS. In2017 International Conference on Computer Science and Engineering (UBMK) 2017 Oct 5 (pp. 816-820). IEEE.

[4] Zhang F, Wang D. An effective feature selection approach for network intrusion detection. In2013 IEEE eighth international conference on networking, architecture and storage 2013 Jul 17 (pp. 307-311). IEEE.

[5] Moustafa N, Slay J. The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Information Security Journal: A Global Perspective. 2016 Apr 4;25:18-31.

[6] Yang R. UAV landmark detection on fast region-based CNN. Arabian Journal of Geosciences. 2021 Jun;14(12):1-9.

[7] Primartha R, Tama BA. Anomaly detection using random forest: A performance revisited. In2017 Inter- national conference on data and software engineering (ICoDSE) 2017 Nov 1 (pp. 1-6). IEEE.

[8] Selvakumar K, Sairamesh L, Kannan A. Wise intrusion detection system using fuzzy rough set- based feature extraction and classification algorithms. International Journal of Operational Research. 2019;35(1):87-107.

[9] Belouch M, El Hadaj S, Idhammad M. A two-stage classifier approach using reptree algorithm for network intrusion detection. International Journal of Advanced Computer Science and Applications. 2017 Jul;8(6):389-94.

[10] Dhanabal L, Shantharajah SP. A study on NSL-KDD dataset for intrusion detection system based on classification algorithms.

Biographies

Dr. S. Brindha is currently working as HoD, Computer Networking Department at PSG Polytechnic College, Coimbatore, TamilNadu. She joined PSG polytechnic College in the year 2000. Her research interestsare in the area of Network. Authentication and she has completed her doctorate in Information and Communication Engineering in the year 2015 from Anna University, Chennai. She has about 24 years of teaching and research experience. Performance Comparison of ASR Models. She has been coordinating the Autonomous Functioning activities for about 16 years. She has published many technical research papers and curriculum design related papers and won Best paper awards in Conferences. She has been instrumental in signing MoU with many companies and setting up industry oriented laboratories.

Ms. A. Dhamayanthi is a Lecturer in the Department of Computer Networking at PSG Polytechnic College, Coimbatore, Tamil Nadu. She joined the institution in 2024 and has two years of teaching experience. Her areas of interest include IoT and Big Data Analytics.

Amish Peniel A (22DC02) is the Students of Diploma in Computer Networking, PSG Polytechnic College.

Harish Raj V (22DC17) is the Students of Diploma in Computer Networking, PSG Polytechnic College.

Hassan A (22DC17) is the Students of Diploma in Computer Networking, PSG Polytechnic College.

Pranith A (22DC37) is the Students of Diploma in Computer Networking, PSG Polytechnic College.

Praveen M (22DC38) is the Students of Diploma in Computer Networking, PSG Polytechnic College.