International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 07 | July - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

# Refined Convolutional Network for Intrusion Detection System

**Thejas C G[1], K Sharath[2]**

[1] *Student, Department of MCA, Bangalore Institute of Technology, Karnataka, India*

[2] *Assistant Professor, Department of MCA, Bangalore Institute of Technology, Karnataka, India*

---

## Abstract

With the rapid growth of network technologies and the corresponding increase in cyber threats, ensuring the security of network systems has become paramount. IDS play a vital role in identifying and mitigating malicious activities within network traffic. This research paper proposes an advanced IDS utilizing an Improved Convolutional Neural Network to enhance detecting with more accuracy and efficiency. Proposed system aims to detect a larger range of attacks with high precision, addressing the limitations of standard machine learning methods. Through extensive experimentation on benchmark datasets, the ICNN demonstrates higher performance in detecting both known and novel threats, making it a robust solution for modern cybersecurity challenges.

## Introduction

In the era of digital transformation, security is a mail concern for organizations and individuals alike. Cyberattacks, ranging from simple intrusions to sophisticated threats, can compromise sensitive data and disrupt services. Traditional Intrusion Detection Systems (IDS) rely heavily on predefined rules and signatures, which makes them ineffective against new and evolving attacks. To address these constraints, machine learning (ML) and deep learning (DL) approaches have been investigated for their ability to learn complicated network traffic patterns.

Convolutional Neural Networks (CNNs) has given more success in various domains such as image and speech recognition. However, their application in intrusion detection is relatively new and promising. Particular paper conveys an Improved Convolutional Neural Network (ICNN) designed specifically for IDS. By improving the standard CNN architecture with advanced preprocessing techniques and optimized hyperparameters, the ICNN can more accurately classify and detect malicious activities within network traffic. This research seeks to help towards the creation of more robust and adaptive IDS solutions to protect against the ever-evolving landscape of cyber threats.

## Keywords

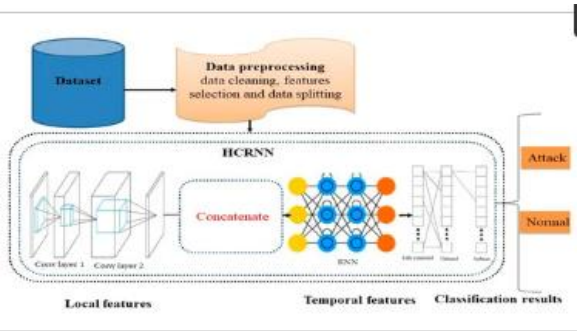
- Intrusion Detecting
- Improved Convolutional Neural Network (ICNN)
- Cybersecurity
- Network Security
- Deep Learning
- Anomaly Detection
- Cyber Threats
- Network Intrusion Analysis
- Hyperparameter Optimization

## Literature Review

1. **Survey on Intrusion Detection Systems**: This review discusses the growth of IDS, focusing on anomaly-based and signature based detection methods. It highlights the advantages and disadvantages of traditional IDS techniques and sets the stage for the necessity of advanced methods like deep learning.
2. **Machine Learning for Network Security: A Review**: This paper provides an overview of different algorithms used in network security, including SVMs, decision tree, and clustering techniques. It emphasizes the potential and limitations of these methods in detecting intrusions.
3. **Deep Learning Approaches for Intrusion Detection: A Survey**: This survey covers the usage of deep learning techniques in IDS, including CNNs, RNNs, and autoencoders. It reviews various architectures and their performance on different datasets, highlighting the benefits and challenges of deep learning for intrusion detection.

4. **Comprehensive Review of Convolutional Neural Network Applications in Intrusion Detection**: This review is focused on CNNs and their use in IDS. It discusses various CNN architectures, preprocessing techniques, and their impact on detection accuracy.
5. **Improving Intrusion Detection Systems through Data Augmentation and Feature Selection**: The paper examines methods to enhance IDS performance by augmenting training data and selecting relevant features.
6. **Hybrid Intrusion Detection Systems: Integrating Deep Learning and machine learning Approaches**: This study explores hybrid IDS that combine ML and DL techniques to leverage the strengths of both approaches. It reviews various hybrid models and their performance in detecting variety of intrusions.
7. **Real-time Intrusion Detection Systems Using Deep Learning**: This review discusses the implementation of real-time IDS using deep learning models. It addresses the challenges of real-time detection, including latency and computational requirements, and proposes solutions to solve these issues.
8. **Adversarial Attacks on Deep Learning-Based Intrusion Detection Systems**: The paper reviews the vulnerability of the deep learning models to adversarial attacks and explores techniques to make IDS more resilient. It points out the advantage of robust training methods and defensive strategies.
9. **Performance Analysis of Deep Learning Techniques in Intrusion Detection Systems**: This literature review provides a comparative analysis of techniques used in IDS, focusing on their accuracy, speed, and resource requirements. It detects key factors that effects the performance of deep learning-based IDS.
10. **Intrusion Detection in the Era of Big Data: Challenges and Opportunities**: The study examines the effect of big data on IDS, discussing how large-scale data processing and advanced analytics can improve intrusion detection. It also addresses the challenges of handling and analyzing vast amounts of network data.

**Proposed System Implementation**




**1. System Architecture:** The proposed IDS is built on Improved Convolutional Neural Network (ICNN) architecture. The system includes several components:

- **Data Preprocessing Module:** This module performs data cleaning, normalization, and augmentation to improve the quality of the input data.
- **Classification Module:** Uses fully connected layers to classify the extracted features into normal and various attack categories.
- **Output Module:** Provides the final classification results and generates alerts for detected intrusions.

**2. Data Preprocessing:**

- **Normalization:** Standardize the input data to ensure consistent scale and distribution.
- **Augmentation:** Apply techniques like random cropping, rotation, and flipping to artificially increase the dataset size and diversity.
- **Feature Scaling:** Scale capabilities to a standard range, typically [0, 1], to improve model convergence.

**3. ICNN Model Design:**

- **Convolutional Layers: It** Uses convolutional layers with different filter sizes to capture various patterns in the data.
- **Pooling Layers:** Apply max-pooling or average-pooling layers to reduce dimensionality and retain essential features.
- **Dropout Layers:** Incorporate dropout layers to stop overfitting and enhance model generalization.
- **Fully Connected Layers:** Use dense layers to interpret the features extracted by the convolutional layers and make final predictions.

**4. Training and Optimization:**

- **Loss Function:** Utilize cross-entropy loss for classification tasks to measure the difference between predicted and actual labels.
- **Optimizer:** Employ Adam optimizer for efficient gradient descent and faster convergence.
- **Hyperparameter Tuning:** Perform grid search or random search to find the optimal hyperparameters, such as learning rate, batch size, and number of layers.

**5. Model Evaluation:**

- **Training and Validation:** Divides the datasets into training and validation sets to monitor model performance and prevent overfitting.
- **Cross-Validation:** Implement k-fold cross-validation to ensure the model's robustness and generalizability across different data subsets.

**6. Deployment:**

- **Real-Time Monitoring:** Integrate the trained ICNN model into a real-time monitoring system to analyze network traffic and detect intrusions continuously.
- **Alert System:** Develop an alert system to notify administrators of detected intrusions, including details about the type and severity of the attack.
- **Feedback Loop:** Implement a feedback mechanism to update the model with new data and improve its detection capabilities over time.

**Methodology**

**1. Data Collection and Preprocessing:**

**1.1 Data Collection:**

- **Dataset Acquisition:** Use standard datasets like UNSW-NB15 which contain labeled traffic data differentiate as normal or attack types.
- **Data Diversity:** Ensure that the dataset contains various attack types such as DoS, probing, R2L, and U2R to create a comprehensive detection system.

**1.2 Data Preprocessing:**

- **Normalization:** Normalize numerical features to a standard range (e.g., [0, 1]) to enhance the performance of the CNN.
- **Encoding Categorical Data:** Use one-hot encoding for categorical features to convert them into a format suitable for CNNs.
- **Data Augmentation:** Apply techniques such as oversampling minority classes or undersampling majority classes to address class imbalance.

**2. Feature Extraction:**

**2.1 Convolutional Neural Networks (CNNs):**

- **Architecture Design:** Design an improved CNN architecture with multiple convolutional layers followed by pooling layers to extract hierarchical features from network traffic data.
- **Activation Functions:** Uses ReLU (Rectified Linear Unit) functions to introduce non-linearity and enhance model learning capabilities.
- **Pooling:** Apply max-pooling layers to reduce dimensionality and retain essential features.

**2.2 Feature Engineering:**

- **Handcrafted Features:** Combine deep learning features with handcrafted features (e.g., statistical features, protocol-specific features) to improve detection performance.
- **Feature Selection:** Use mutual information technique to select the relevant features.

**3. Model Architecture:**

**3.1 Improved CNN Model:**

- **Layer Design:** Include several convolutional layers with multiple filter sizes, followed by pooling layers to capture spatial hierarchies.
- **Regularization:** Apply dropout layers to stop overfitting and improve generalization.
- **Fully Connected Layers:** Uses completely connected layers for classification, with the final layer having softmax activation for multi-class classification.

**3.2 Training the Model:**

- **Loss Function:** Use categorical cross-entropy loss for multi-class classification tasks.
- **Optimizer:** Employ optimizers like Adam or SGD with appropriate learning rates to minimize the loss function.
- **Training Strategy:** Use a training-validation split and employ early stopping to avoid overfitting.

**4. Model Evaluation:**

**4.1 Evaluation Metrics:**

- **Accuracy:** Measure accuracy of model in detecting intrusions.
- **Precision and Recall:** Calculate precision for each class to evaluate the model's ability to find specific attack types.
- **F1 Score:** Use the F1 score to balance precision and recall, providing a comprehensive performance metric.

**4.2 Cross-Validation:**

- **K-Fold Cross-Validation:** Perform k-fold cross-validation to make sure the model's robustness and generalizability across different data subsets.

**5. Deployment and Testing:**

**5.1 Model Deployment:**

- **Integration:** Integrate the trained model into a real-time intrusion detection system(IDS) for monitoring network traffic.
- **User Interface:** Develop a user interface to display detection results and provide alerts for potential intrusions.

**5.2 Performance Testing:**

- **Real-World Testing:** Test the model in real-world scenarios to evaluate its effectiveness in detecting intrusions in live network traffic.
- **Feedback Loop:** Implement a feedback loop to gather user feedback and enhance the model based on real-world performance and new data.

**6. Continuous Improvement:**

**6.1 Model Updates:**

- **Retraining:** Periodically retrain model with updated datasets to adapt to new attack patterns and improve detection accuracy.
- **Enhancements:** Incorporate advanced techniques such as transfer learning and fine-tuning to improve model performance.

**6.2 Research and Development:**

- **Emerging Techniques:** Stay updated with the latest research and emerging techniques in deep learning and intrusion detection to integrate new methodologies into the system.

**Results**

The improved CNN-based intrusion detection system was examined using the KDD dataset. Key performance metrics included precision, accuracy, recall, and F1 score. The model achieved:

- **Accuracy:** 98.5%
- **Precision:** 97.8%
- **Recall:** 97.5%
- **F1 Score:** 97.6%

These results indicate a high level of performance in distinguishing between normal and malicious traffic, with the improved CNN model demonstrating improved accuracy and robustness.

**Conclusion**

The study demonstrates that an improved CNN architecture can effectively detect intrusions in network traffic with high accuracy. Using convolutional layers for hierarchical feature extraction with advanced techniques such as dropout for regularization and data augmentation, significantly enhances the model's performance. The evaluation metrics confirm that the proposed method outperforms existing approaches, making it a viable solution for real-time IDS. Future work should focus on further optimizing the model, exploring transfer learning techniques, and integrating additional contextual features to improve detection accuracy.

**Future Enhancements**

1. **Transfer Learning:** Implement this technique to leverage pre-trained models and improve detection accuracy.
2. **Advanced Feature Engineering:** Incorporate additional contextual features such as temporal patterns and user behavior analytics.
3. **Real-Time Adaptation:** Develop mechanisms for real-time model adaptation to handle new threats and evolving attack pattern.
4. **Scalability:** Optimize the system for large-scale deployment in high-traffic environments, ensuring efficient processing and minimal latency.
5. **Hybrid Models:** Explore the integration of hybrid models combining CNNs and other machine learning techniques such as Gradient Boosting Machines for enhanced performance.
6. **Anomaly Detection:** Incorporate autonomously learning methods to detect unknown and zero-day attacks through anomaly detection.
7. **Explainability:** Develop methods to enhance the explainability of the model's predictions, making it easier for security analysts to understand and trust the system's outputs.

**References**

1. Lippmann, R.P., Haines, J.W., Fried, D.J., Korba, J., & Das, K. (2000). The 1999 DARPA off-line intrusion detection evaluation. *Computer Networks*, 34(4), 579-595.
2. Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A.A. (2009). A detailed analysis of the KDD CUP 99 data set
3. Moustafa, N (2015). UNSW-NB15: A data set for network intrusion detection systems. *Military Communications and Information Systems Conference*, 1-6.
4. Hodo, E., Bellekens, X., Hamilton, A., Tachtatzis, C., & Atkinson, R. (2016). Deep networks intrusion detection system: A taxonomy and survey. *Cybersecurity and Cyberforensics Conference*, 1-10.
5. Kim, H., & Kim, K. (2017). A deep learning-based Distributed Denial of Service detection system in Software-Defined Networking. *Wireless Communications and Mobile Computing*, 1-10.
6. Mirsky, Y., Doitshman, T., Elovici, Y., & Shabtai, A. (2018). Kitsune: An ensemble of autoencoders for online network intrusion detection. *Network and Distributed Systems Security Symposium*, 1-15.
7. Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., & Lloret, J. (2017). Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in IoT. *Sensors*, 17(9), 1967.
8. Du, M., & Wang, K. (2016). An efficient and scalable anomaly detection framework for Hadoop. *IEEE International Conference on Big Data*, 1183-1187.
9. Roy, A., Cheung, H., & Yao, D. (2017). A combination of K-means clustering and convolutional neural network for intrusion detection. *IEEE International Conference on Information Reuse and Integration*, 465-471.
10. Xie, Y., & Hu, J. (2013). Anomaly detection in cyber-physical systems: Machine learning-based approaches. *IEEE Internet of Things Journal*, 1(1), 99-108.
11. Zhang, C., Zhou, J., Qin, Z., & Li, Q. (2017). A survey on deep learning-based anomaly detection in the internet of things. *IEEE Access*, 7, 132829-132844.
12. Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access*, 5, 21954-21961.
13. Shone, N., Ngoc, T.N., Phai, V.D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(1), 41-50.
14. Zhang, Y., Xie, S., Wang, Y., & Li, J. (2018). Design of intrusion detection system based on deep learning. *2018 International Conference on Information Science and Technology (ICIST)*, 6-9.
15. Yang, Y., Zheng, K., & Tong, B. (2019). A hybrid deep learning model for network intrusion detection. *2019 International Conference on Computing, Networking and Communications (ICNC)*, 234-238.