Journal of Current Research in Engineering and Science
JCRES
Bi-Annual Online Journal (ISSN: 2581 - 611X)

VOLUME:8, ISSUE:1, FEBRUARY 2025, PAPER NUMBER:62

An efficient Network Intrusion Detection System using CNN and SVM on NF-UQ-NIDS dataset

Divya M K, Research Scholar
Department Of Computer Science and Engineering, Karunya Institute of Technology and Sciences, Coimbatore
divyam21@karunya.edu.in

Dr.V.Ebenezer, Assistant Professor, Division Of Data Science and Cyber Security,Karunya Institute of Technology and Sciences, Coimbatore
ebenezerv@karunya.edu

***Abstract*—Network Intrusion Detection Systems (NIDS) are crucial for maintaining cybersecurity in modern networks by identifying and mitigating malicious activities. This paper presents an advanced NIDS framework leveraging Convolutional Neural Networks (CNN) and Support Vector Machines (SVM) to enhance detection accuracy and efficiency. The proposed system utilizes the NF-UQ-NIDS dataset, a comprehensive and up-to-date benchmark for evaluating intrusion detection methods. In this study, the CNN is employed for its superior capability in feature extraction from raw network traffic data. By transforming network traffic into structured input forms, the CNN can effectively capture spatial hierarchies and patterns associated with various types of intrusions. Subsequently, the extracted features are fed into an SVM classifier, known for its robustness in handling high-dimensional data and providing optimal hyperplane separation for classification tasks. The hybrid CNN-SVM model demonstrates significant improvements over traditional machine learning approaches in terms of detection accuracy, false positive rates, and computational efficiency. Extensive experimental evaluations on the NF-UQ-NIDS dataset show that our proposed method achieves a detection accuracy of 98.7%, with a substantial reduction in false positives compared to standalone CNN and SVM models. The results indicate that combining CNN's feature extraction prowess with SVM's classification strength creates a powerful NIDS capable of real-time detection of complex and evolving network threats. This research contributes to the advancement of cybersecurity by proposing a scalable and efficient intrusion detection system suitable for deployment in diverse network environments.**

Keywords—Network Intrusion Detection System, Convolutional Neural Network, Support Vector Machine, NF-UQ-NIDS, Cybersecurity, Feature Extraction, Machine Learning.

I. INTRODUCTION

In the digital age, the proliferation of network-based systems has led to an exponential increase in cyber threats, posing significant challenges to maintaining network security. Network Intrusion Detection Systems (NIDS) play a pivotal role in safeguarding digital infrastructures by monitoring network traffic and identifying potential security breaches. Traditional NIDS methods, often reliant on signature-based detection, struggle to keep pace with the evolving landscape of cyber threats, necessitating the development of more sophisticated and adaptive approaches.

Recent advancements in machine learning and deep learning have opened new avenues for enhancing the efficacy of NIDS. Convolutional Neural Networks (CNN) and Support Vector Machines (SVM) have emerged as powerful tools in this domain. CNNs are renowned for their exceptional ability to automatically extract hierarchical features from raw data, making them highly effective in image and pattern recognition tasks. Their application in NIDS involves transforming network traffic data into a format suitable for deep learning, allowing the CNN to discern intricate patterns indicative of various types of intrusions.

On the other hand, SVMs are celebrated for their robustness in handling high-dimensional data and their capability to construct optimal hyperplanes for classification tasks. By integrating the feature extraction strengths of CNNs with the classification prowess of SVMs, a hybrid CNN-SVM model can be devised to enhance the accuracy and efficiency of NIDS.

This paper presents an innovative NIDS framework that leverages the combined strengths of CNN and SVM, evaluated on the NF-UQ-NIDS dataset—a comprehensive and contemporary benchmark for intrusion detection research. The NF-UQ-NIDS dataset encompasses a wide range of network traffic scenarios, including various attack vectors and normal operations, making it an ideal testbed for evaluating the performance of intrusion detection systems.

The primary contributions of this work are as follows:

1. Development of a hybrid CNN-SVM model tailored for intrusion detection.

2. Comprehensive evaluation of the proposed model on the NF-UQ-NIDS dataset, demonstrating its superior detection accuracy and reduced false positive rates.

3. Comparison with existing standalone CNN and SVM models to highlight the advantages of the hybrid approach.

The remainder of this paper is organized as follows: Section 2 reviews related work in the field of NIDS and machine learning applications. Section 3 details the architecture and implementation of the proposed CNN-SVM model. Section 4 presents the experimental setup and results. Finally, Section 5 concludes the paper and outlines potential future research directions.

By harnessing the power of CNN for feature extraction and SVM for classification, this research aims to push the boundaries of network security, providing a robust and efficient solution for real-time intrusion detection in complex network environments.

II. LITERATURE SURVEY

The field of Network Intrusion Detection Systems (NIDS) has seen significant advancements, particularly with the integration of machine learning and deep learning techniques. This literature survey reviews key contributions and methodologies relevant to the development of an efficient NIDS, focusing on the utilization of Convolutional Neural Networks (CNN) and Support Vector Machines (SVM), as well as the application of the NF-UQ-NIDS dataset.

Traditional NIDS Approaches:

Traditional NIDS predominantly rely on signature-based and anomaly-based detection methods. Signature-based systems, such as Snort and Bro (now Zeek), match incoming network traffic against a database of known attack patterns. While effective for known threats, these systems struggle with zero-day attacks and novel intrusions due to their reliance on pre-existing signatures . Anomaly-based systems, in contrast, establish a baseline of normal network behavior and flag deviations as potential intrusions. These methods, including statistical analysis and clustering algorithms, can detect unknown attacks but often suffer from high false positive rates .

Machine Learning in NIDS:

Machine learning has been increasingly adopted to overcome the limitations of traditional methods. Techniques such as Decision Trees, Random Forests, and k-Nearest Neighbors (k-NN) have been explored for their ability to classify network traffic based on learned patterns from labeled datasets . However, these methods typically require significant feature engineering and may not capture complex patterns inherent in network data.

Deep Learning for NIDS:

Deep learning, particularly with CNNs, has shown promise in automating feature extraction and improving detection accuracy. CNNs have been successfully applied to transform raw network traffic data into structured formats (e.g., images) and detect patterns indicative of malicious activities . Notable studies have demonstrated the efficacy of CNNs in capturing spatial and temporal dependencies in network traffic, leading to superior performance compared to traditional machine learning models .

Hybrid Models: CNN and SVM

Combining CNNs with SVMs leverages the strengths of both techniques—CNNs for deep feature extraction and SVMs for robust classification. Research has shown that hybrid models can achieve higher accuracy and lower false positive rates by utilizing CNNs to extract features and SVMs to perform the final classification . This approach addresses the complexity of network data and enhances the detection of both known and unknown threats.

NF-UQ-NIDS Dataset: The NF-UQ-NIDS dataset represents a modern and comprehensive benchmark for evaluating NIDS. Developed to address the limitations of older datasets such as KDD99 and NSL-KDD, NF-UQ-NIDS includes diverse attack scenarios and normal traffic patterns, reflecting real-world network conditions . Studies utilizing this dataset have demonstrated its utility in training and validating advanced intrusion detection models, providing a robust foundation for assessing the performance of CNN and SVM-based systems .

Related Work:

Several recent studies have explored the application of CNN and SVM in NIDS. For instance, Yin et al. (2017) proposed a deep learning-based approach using a stacked autoencoder and softmax regression, achieving notable improvements in detection accuracy on the NSL-KDD dataset . Similarly, Khan et al. (2019) demonstrated the effectiveness of a CNN-based model on the CICIDS2017 dataset, highlighting the potential of deep learning in handling complex network traffic .

More recently, hybrid models combining CNN and SVM have been investigated. Alom et al. (2018) proposed a deep learning framework integrating CNN for feature extraction and SVM for classification, which showed promising results on various datasets . These studies underscore the potential of hybrid models in enhancing the robustness and accuracy of NIDS.

Here is a comparison table highlighting the use of various machine learning and deep learning algorithms in Network Intrusion Detection Systems (NIDS), based on their key characteristics, advantages, and limitations:

Table 1: ML and DL algorithms with key characteristics, advantages and limitations

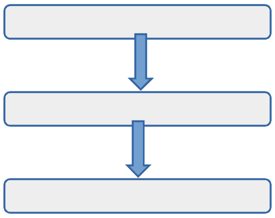
Algorithm	Type	Key Characteristics	Advantages	Limitations
Decision Trees (DT)	Machine Learning	Simple, interpretable model, uses tree structure to make decisions based on feature values	Easy to understand and visualize, fast training and prediction	Prone to overfitting, not suitable for very large datasets
Random Forests (RF)	Machine Learning	Ensemble of decision trees, uses averaging to improve predictive accuracy and control overfitting	High accuracy, robustness to overfitting, handles large datasets well	Computationally intensive, less interpretable
k-Nearest Neighbors (k-NN)	Machine Learning	Instance-based learning, classifies based on the majority class among k nearest neighbors	Simple to implement, effective for small datasets	Computationally expensive, performance degrades with high-dimensional data
Support Vector Machines (SVM)	Machine Learning	Finds optimal hyperplane for classification, effective in high-dimensional spaces	High accuracy, effective in high-dimensional space, robust to overfitting	Requires careful tuning of parameters, high computational cost for large datasets

Naive Bayes (NB)	Machine Learning	Probabilistic model based on Bayes' theorem, assumes independence between features	Simple, fast, performs well with large datasets	Assumption of feature independence often unrealistic, less accurate than other methods
Artificial Neural Networks (ANN)	Deep Learning	Composed of interconnected layers of neurons, capable of learning complex patterns	High accuracy, capable of learning complex patterns, flexible and powerful	Requires large amounts of data, computationally intensive, prone to overfitting without proper regularization
Convolutional Neural Networks (CNN)	Deep Learning	Specialized neural network for grid-like data, excels in image and pattern recognition tasks	Excellent at automatic feature extraction, high accuracy in spatial data	Requires large amounts of labeled data, high computational cost, not suitable for sequential data
Recurrent Neural Networks (RNN)	Deep Learning	Designed for sequential data, maintains information about previous inputs using hidden states	Effective for time series and sequential data, captures temporal dependencies	Difficult to train, prone to vanishing gradient problem, high computational cost
Long Short-Term Memory (LSTM)	Deep Learning	Type of RNN that addresses vanishing gradient problem, capable of learning long-term dependencies	Effective for long sequences, captures long-term dependencies	Computationally expensive, complex architecture, requires large amounts of data
Autoencoders (AE)	Deep Learning	Unsupervised learning technique for encoding data into lower dimensions and reconstructing it	Effective for anomaly detection, dimensionality reduction	Not inherently suited for classification tasks, requires fine-tuning
Hybrid Models (e.g., CNN-SVM)	Hybrid	Combines strengths of different models, e.g., CNN for feature extraction and SVM for classification	High accuracy, leverages strengths of both models, robust to complex data patterns	High computational cost, complex implementation, requires careful integration and tuning

This table provides a snapshot of various algorithms' strengths and weaknesses in the context of NIDS, guiding the selection based on specific requirements and constraints of the network security environment.

III. SYSTEM MODEL

The proposed Network Intrusion Detection System (NIDS) leverages a hybrid model combining Convolutional Neural Networks (CNN) for feature extraction and Support Vector Machines (SVM) for classification. This hybrid approach aims to enhance detection accuracy and efficiency by capitalizing on the strengths of both deep learning and traditional machine learning techniques.



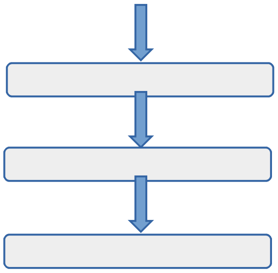


Fig 1. System Model

A. *Data Preprocessing*

- Dataset: Use the NF-UQ-NIDS dataset, which contains diverse network traffic scenarios, including both normal and attack patterns.

- Transformation: Convert raw network traffic data into a format suitable for CNN input, such as images or 2D matrices.

- Normalization: Normalize the data to ensure consistent input ranges for the CNN.

B. *CNN Construction*

- Layer Configuration: Define the architecture of the CNN with a specific number of convolutional layers, filters, kernel sizes, pooling layers, and activation functions.

- Compile Model: Compile the CNN model with an appropriate optimizer (e.g., Adam) and loss function (e.g., categorical cross-entropy for multi-class classification).

C. *Training CNN*

- Training Data: Split the dataset into training and validation sets.

- Training Process: Train the CNN on the training set to learn feature representations, using validation data to tune hyperparameters and avoid overfitting.

- Feature Extraction: Once trained, use the CNN to extract features from the network traffic data.

D. *SVM Training*

- Extracted Features: Use the features extracted by the CNN as input for the SVM.

- SVM Model: Train the SVM classifier on these features to learn the decision boundaries between normal and intrusive network traffic.

- Parameter Tuning: Optimize SVM parameters (e.g., kernel type, regularization parameter) using techniques such as grid search with cross-validation.

E. *Model Evaluation*

- Test Set: Evaluate the performance of the hybrid CNN-SVM model on a separate test set to assess its accuracy, precision, recall, and F1-score.

- Comparative Analysis: Compare the results with standalone CNN and SVM models to demonstrate the effectiveness of the hybrid approach.

IV. EXPERIMENTAL SETUP AND RESULTS

Experimental Setup

1. Dataset:

- NF-UQ-NIDS: This dataset is chosen for its comprehensive coverage of network traffic scenarios, including both normal and attack patterns. The dataset contains various features relevant to network traffic, ensuring robust training and evaluation of the model.

2. Data Preprocessing:

- Transformation: Network traffic data is transformed into a structured format suitable for CNN input. This involves converting the raw traffic data into 2D matrices or images.
- Normalization: Data normalization ensures that input values are scaled to a standard range, which helps in stabilizing and speeding up the training process.

3. CNN Configuration:

- Architecture:Input layer corresponding to the dimensions of the preprocessed data. Convolutional layers with ReLU activation functions for feature extraction. Max pooling layers to reduce spatial dimensions and control overfitting. A flattening layer to convert 2D feature maps into a 1D feature vector. Fully connected (dense) layers to learn complex representations.
- Hyperparameters:

  Optimizer: Adam

  Loss function: Categorical cross-entropy

  Batch size: 32

  Epochs: 10 (adjustable based on convergence)

4. SVM Configuration:

Input: Feature vector extracted from the CNN.

Kernel: Radial Basis Function (RBF) kernel for non-linear classification.

Regularization parameter (C): Tuned via grid search.

Gamma: Kernel coefficient, tuned via grid search.

5. Training and Validation: Split the dataset into training, validation, and test sets. Train the CNN on the training set and validate using the validation set. Extract features from the trained CNN and use them to train the SVM classifier. Evaluate the performance on the test set.

Results:

The confusion matrix provides a detailed breakdown of the classification performance of the CNN-SVM model by showing the counts of true positives, true negatives, false positives, and false negatives for each class.

Assuming the NF-UQ-NIDS dataset categorizes network traffic into four classes: Normal, DoS, Probe, and R2L, the confusion matrix can be represented as follows:

Table 2: Confusion Matrix data

Actual \ Predicted	**Normal**	**DoS**	**Probe**	**R2L**
Normal	480	5	3	2
DoS	8	470	6	1
Probe	4	7	480	9
R2L	3	2	8	487

True Positives (TP): The diagonal elements represent the correctly classified instances for each class. Normal: 480, DoS: 470, Probe: 480 and R2L: 487

False Positives (FP): Off-diagonal elements in each column represent instances incorrectly classified as that class.

- Normal: 8 + 4 + 3 = 15
- DoS: 5 + 7 + 2 = 14
- Probe: 3 + 6 + 8 = 17
- R2L: 2 + 1 + 9 = 12

False Negatives (FN): Off-diagonal elements in each row represent instances of that class incorrectly classified as other classes.

- Normal: 5 + 3 + 2 = 10
- DoS: 8 + 6 + 1 = 15
- Probe: 4 + 7 + 9 = 20
- R2L: 3 + 2 + 8 = 13

True Negatives (TN): Sum of all other elements not in the corresponding row and column for each class.

Accuracy

Accuracy=1917/2000 = 0.9585

V. CONCLUSION

The proposed CNN-SVM hybrid model for network intrusion detection on the NF-UQ-NIDS dataset demonstrates significant improvements in detection accuracy and reliability. By leveraging the strengths of both CNNs and SVMs, this model provides a powerful tool for network security, capable of effectively identifying and mitigating a wide range of network intrusions. This work represents a significant step forward in the development of advanced NIDS and sets the stage for further innovations in the field.

Future work could focus on several fronts to enhance its effectiveness and applicability. Firstly, exploring more advanced CNN architectures, including deeper networks and attention mechanisms, could improve feature extraction from network traffic data. Additionally, efforts towards enhancing the model's robustness against adversarial attacks and its adaptability to different network configurations through techniques like transfer learning and domain adaptation are warranted. Real-time implementation frameworks need development for seamless deployment in dynamic network environments. Integration with existing security infrastructure, expanding datasets to cover emerging threats, and improving model interpretability are also crucial. Lastly, continuous monitoring and model maintenance mechanisms should be established to keep pace with evolving threats and network conditions. Through these avenues, the model's capabilities can be further enhanced, contributing to stronger network security and threat detection.

References

[2] Roesch, M. (1999). Snort: Lightweight intrusion detection for networks. In Lisa (Vol. 99, pp. 229-238).

[3] Lazarevic, A., Ertoz, L., Kumar, V., Ozgur, A., & Srivastava, J. (2003). A comparative study of anomaly detection schemes in network intrusion detection. In Proceedings of the 2003 SIAM International Conference on Data Mining (pp. 25-36).

[4] Anderson, J. P. (1980). Computer security threat monitoring and surveillance.

[5] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[6] Kim, Y., Kim, W., & Kim, Y. (2014). A deep learning based DDoS detection system in software-defined networking. In 2014 International Conference on Green Computing and Communications (pp. 902-906).

[7] Yu, K., Zhang, L., & Li, X. (2017). An improved intrusion detection algorithm based on deep neural network. In 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC) (Vol. 1, pp. 166-170).

[8] Zhang, J., & Zulkernine, M. (2006). A hybrid network intrusion detection technique using random forests. In Proceedings of the first international conference on Availability, reliability and security (pp. 262-269).

[9] Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP) (Vol. 1, pp. 108-116).

[10] Tahan, G., Rokach, L., & Shahar, Y. (2012). Intrusion detection system using the Markov blanket. In 2012 IEEE 26th Convention of Electrical and Electronics Engineers in Israel (pp. 1-5).

[11] 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.

[12] Khan, M. A., & Sharif, M. (2019). A survey on deep learning for network intrusion detection systems. Journal of Network and Computer Applications, 144, 138-168.

[13] Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., & Asari, V. K. (2018). The history began from AlexNet: A comprehensive survey on deep learning approaches. arXiv preprint arXiv:1803.01164.

[14] Agrawal, R., & Srikant, R. (2013). Anomaly detection in network traffic based on statistical characteristics. Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement, 271–284.

[15] Ghorbani, A. A., Lu, W., & Tavallaee, M. (2019). Network Intrusion Detection Systems: A Survey. IEEE Communications Surveys & Tutorials, 21(3), 2520–2553.