Deep Learning-based Frameworks for Real-time Cyber Threat Analysis.

Laila Almutairi.

Department of Computer Engineering, College of Computer and Information Sciences,
Majmaah University, Al-Majmaah, 11952, Saudi Arabia, l.almutairi@mu.edu.sa

Abstract

This study attempts to enhance cybersecurity defenses through the development of robust deep learning frameworks for real-time cyber threat analysis, focusing on the effectiveness of convolutional neural networks (CNNs) in the domain of dynamic cybersecurity. The primary aim is to swiftly detect, classify, and mitigate emerging cyber threats in network environments. We assess the proposed convolution neural network-based intrusion detection (CNNID) framework using the "NSL-KDD" dataset from the Canadian Institute for Cybersecurity at the University of New Brunswick as a benchmark. We recognized a potential bias in selecting TensorFlow for evaluation, underscoring the need to discuss limitations and compare them with alternative deep learning frameworks, thereby improving the generalizability of the findings. This research contributes to the discourse on utilizing diverse deep learning methods for proactive cyber threat detection, presenting metrics including a false acceptance rate of 0.88, F1-score of 98.50%, recall of 97.15%, precision of 99.99%, and accuracy of 97.10%.

Keywords:

Deep Learning; Frameworks; Real-time; Cyber Threat Analysis.

1. Introduction

In this era of digital interconnectivity, the omnipresent risk of cyber-attacks significantly challenges the protection and integrity of confidential information [1]. As both organizations and individuals increasingly utilize technological advancements, the imperative of comprehensive cybersecurity measures is paramount [2]. Conventional methods for cyber threat analysis frequently lag in addressing the rapidly evolving and sophisticated nature of these threats, necessitating a paradigm shift toward more innovative strategies [3].

Deep learning (DL)—a subset of machine learning—has emerged as a potent tool in addressing the dynamic spectrum of cyber threats [4]. This paper aims to delve into the domain of real-time cyber threat analysis, with a particular emphasis on examining the effectiveness of convolutional neural networks (CNNs), which is a specialized category of deep neural networks[5].

Research Question: How effective are CNN-based frameworks in real-time cyber threat analysis considering factors such as adaptability to evolving threats, interpretability, computational efficiency, and scalability to varying network environments?

CNNs have exhibited remarkable effectiveness in various domains, such as image recognition and natural language processing, rendering their application in cybersecurity a promising avenue [6]. The impetus for employing DL in this context is its capacity to discern complex patterns and features autonomously from extensive datasets, thereby enabling more nuanced and adaptable threat detection [7].

With the rapid evolution of the digital landscape, the ability to promptly detect, classify, and address cyber threats in real time has become crucial for strengthening cybersecurity defenses [8]. This study employs the widely acknowledged "NSL-KDD dataset" from the Canadian Institute for Cybersecurity at the University of New Brunswick to thoroughly evaluate the performance of CNN-based frameworks [9].

The selection of "TensorFlow" as the evaluation framework is strategic, owing to its well-established robustness in managing deep neural networks and its widespread adoption within the research and development community [11]. This research aims to evaluate the effectiveness of CNN-based frameworks in real-time cyber threat analysis, as well as explore critical aspects such as model interpretability, computational efficiency, and adaptability to different network environments. These elements are essential for the practical implementation of such frameworks [13][14].

Through an exhaustive analysis of the strengths and weaknesses of the proposed approach, this study aims to offer significant contributions to the existing dialogue on employing deep learning methodologies for proactive and efficient cyber threat detection and response mechanisms [15].

The novel contributions of this study are as follows:

1. Dynamic Learning: Introducing adaptive threat detection using CNNs for real-time analysis, which allows the model to evolve alongside emerging cyber threats.
2. Behavioral Insight: Incorporating behavioral analysis into CNN frameworks to enhance the precise identification of subtle anomalies in network activities.
3. Scalability Optimization: Proposing a multiclass network intrusion detection approach for real-time scalability by refining the CNN architecture for efficient computation across various network environments.
4. Enhanced Interpretability: Developing methods to augment the interpretability of CNN-based models, thus providing deeper insights into their decision-making processes for more informed response strategies.

2. Literature Review

In 2022, Otoum et al. [16] developed an advanced intrusion detection system (IDS) employing deep learning (DL), DL-IDS, for identifying security threats in IoT contexts. Although numerous IDSs exist, several lack optimal features, learning capabilities, and data management practices that are crucial for precise attack detection. DL-IDS utilizes the spider monkey optimization (SMO) method for feature selection and the stacked-deep polynomial network (SDPN) for distinguishing between normal and abnormal data. The integration of these two algorithms in DL-IDS enables superior detection and identification capabilities, effectively identifying abnormalities such as R2L (remote-to-local), U2R (user-to-root), and DoS (denial of service) attacks. Comparative analyses reveal that DL-IDS surpasses existing systems in F-score, recall, accuracy, and precision metrics.

Jothi & Pushpalatha (2023) [17] introduced an innovative IDS for IoT networks, employing Whale Integrated LSTM (WILS) networks. This system is adept at profiling standard IoT device performance, detecting malicious devices during attacks, and predicting attack types. The IDS was tested in real-time IoT network scenarios simulated using the OMNET-python API, and benchmarked using prominent datasets like CIDDS-001 and UNSWNB15. Extensive testing demonstrated the superiority of the WILS models over existing models in recall, precision, and accuracy, thereby confirming their effectiveness in bolstering IoT network security.

Ferrag et al. (2021) [18] conducted a comprehensive study on federated deep learning methods for IoT cybersecurity, encompassing a broad spectrum of IoT sectors, including industrial IoT and the internet of vehicles. The research encompassed an examination of combined learning-based security solutions, blockchain integration, malware detection in IoT applications, and vulnerability identification. The authors performed an experimental analysis under both centralized and federated learning scenarios, comparing RNNs, CNNs, and DNNs. The study demonstrated that federated deep learning enhances attack detection accuracy and ensures data privacy from IoT devices more effectively than traditional non-federated learning using real-world datasets such as Bot-IoT, MQTTset, and TON_IoT.

In 2020, Sriram et al. [19] developed a “botnet detection system” using DL, which is designed to operate on network traffic flow. This framework functions by collecting flow data, converting them into connection records, and employing a deep learning model to identify attacks originating from compromised IoT devices. Extensive testing was conducted on widely recognized and published benchmark datasets to determine the most effective deep-learning model. Additionally, these datasets were visually presented to enhance the understanding of their characteristics. The DL model demonstrated superior performance compared to traditional machine learning (ML) models.

Aslan et al. [20] innovatively combined a hybrid architecture using two extensive pre-trained network models. The process comprised four main stages: data collection, planning of the deep neural network architecture, training, and performance evaluation. The proposed technique was tested using datasets such as Malevis, Microsoft BIG 2015, and Malimg. The experimental results indicated that this approach surpassed existing methods in malware classification literature, achieving a notable 97.78% accuracy on the Malimg dataset, thereby outperforming other ML-based malware detection methods.

The 2021 survey by Lansky et al. [21] extensively reviewed and categorized intrusion detection systems (IDSs) based on deep learning. The study started with an introduction to the fundamental concepts of IDS architecture and various deep learning methods. Thereafter, these methods were categorized based on the type of deep learning employed. This paper detailed the application of deep learning networks for precise intrusion detection in IDS. The analysis concluded with a thorough review of the examined IDS frameworks, offering final observations and highlighting potential avenues for future research.

In 2022, Amanullah et al. [22] compiled a comprehensive review of the latest advancements in big data, IoT security, and deep learning. Their work included an analysis of the interconnections and comparisons between deep learning, big data, and IoT security technologies. Furthermore, they developed a thematic taxonomy by comparing technical work across these three domains. The study concluded with an overview of the challenges in integrating big data technologies with deep learning for IoT security and provided directions for future research in this field.

Javeed et al. [23] (2021) developed a novel approach to combating cyberattacks and threats through the creation of software-defined networks (SDNs), which synergize programmability with centralized intelligence. This model facilitates a cohesive and efficient security solution. They introduced an "SDN-enabled architecture" to reduce the burden on the limited resources of IoT devices, utilizing hybrid deep learning detection algorithms for the efficient identification of cyber assaults and threats. The cutting-edge CICDDoS 2019 dataset was used to train the system. The algorithm demonstrated high accuracy, reduced testing time, and a low false positive rate (FPR). The findings were validated against existing benchmark algorithms and confirmed through 10-fold cross-validation to ensure objectivity.

In their 2021 research, Lee et al. [24] emphasized cybersecurity as a paramount concern in enterprise risk management, especially as businesses accelerate digital transformation through IoT services, cloud computing, social media, and mobile devices. The study highlighted the complexities of cybersecurity management in the face of evolving data protection and privacy regulations. A cyber risk management framework was presented, detailing a cyber risk assessment process and utilizing a real-world cybersecurity case to illustrate the relationship between cyber investment cost analysis and the continuous improvement of cybersecurity performance. This was contextualized against the backdrop of emerging cybersecurity technologies and the rapidly increasing number of cyber breaches.

In 2019, Venkatraman et al. [25] developed an innovative method for malware detection that ingeniously integrates deep learning with visualization techniques. They explored the use of image-based methods to detect "suspicious system activity" and investigated the application of hybrid image-based methods with DL architectures for effective malware classification. The performance of these methods can be evaluated using cost-sensitive deep learning architectures and various similarity metrics for analyzing malware behavior patterns. The hybrid approach was tested using both publicly available and privately collected extensive malware datasets to assess scalability. The results confirmed the high accuracy of the developed malware classifiers.

In 2022, Haq et al. [44] designed an intrusion detection solution using a CNN tailored for enhanced data rates for GSM evolution (EDGE) computing. The system differentiated between attack and non-attack events, utilizing both multiclass and binary classifications. In the binary classification, all malicious traffic data were aggregated into a single category. A 13-layer sequential 1-D CNN was implemented for the IDS, and its performance was evaluated using the NSL-KDD dataset. Principal component analysis (PCA) was employed to reduce the feature vector size through feature extraction and engineering. The developed principal component-based CNN (PCCNN) achieved remarkable accuracies of 99.34% for binary classification and 99.13% for multiclass classification on the NSL-KDD dataset. These experimental results underscore the effectiveness of the PCCNN approach, highlighting its potential for deep learning-based intrusion detection in IoT systems.

Table 1. Summary of Research Gaps

Ref No	.s/Year	Method	Finding	Research Gap
[16]	Otoum et al. (2022)	DL-IDS with SMO and SDPN	Optimal detection in IoT, Improved accuracy, precision, recall, and F-score	Lack of optimal features learning and data set management in existing IDSs
[17]	Jothi & Pushpalatha (2023)	WILS Networks	Profiling IoT device performance, Improved accuracy, precision, and recall	Limited focus on advanced IDS for IoT networks
[18]	Ferrag et al. (2021)	Federated Deep Learning	Outperformed RNN, CNN, and DNN, Improved data privacy	Lack of federated learning-based security systems in IoT cybersecurity
[19]	Sriram et al. (2020)	DL Botnet Detection	Outperformed conventional ML models	Limited focus on DL-based botnet detection using network traffic flows
[20]	Aslan et al. (2021)	Hybrid Architecture	High accuracy in malware classification	Limited exploration of hybrid architectures in malware detection
[21]	Lansky et al. (2021)	Survey and Classification	In-depth analysis of deep learning-based IDS, Classification of schemes	Need for comprehensive survey on deep learning-based intrusion detection schemes
[22]	Amanullah et al. (2022)	Thematic Taxonomy	Comparative analysis of DL, IoT security, and big data technologies	Lack of thematic taxonomy and comprehensive survey
[23]	Javeed et al. (2021)	SDN-enabled Architecture	Efficient detection of cyber threats, Minimal false positive rate	Integration of SDN with hybrid deep learning for cyber threat detection

[24]	Lee et al. (2021)	Cyber Risk Management Framework	Continuous improvement of cybersecurity performance	Importance of cyber risk management in the context of digital transformation
[25]	Venkatraman et al. (2019)	Hybrid Deep Learning and Visualization	Effective malware detection, High accuracy	Application of image-based approaches in malware classification
[44]	Haq et al. (2022)	CNN for EDGE Computing	Accuracies: 99.34% (Binary) / 99.13% (Multiclass)	Potential for IoT intrusion detection using PCCNN

Despite progress in DL-based IDS and cybersecurity frameworks, there remains a significant research gap in optimal feature learning and dataset management. The efficiency of several existing IDSs is compromised in these critical areas, affecting the accuracy of attack detection. Additionally, the scalability and adaptability of these frameworks in dynamic network environments require further exploration. The literature indicates some advancements, but a substantial gap persists in understanding and addressing these fundamental challenges, prompting the need for more focused research in this domain.

2.1 Problem Statement

The field of cybersecurity confronts a significant challenge: achieving optimal feature learning and effective dataset management within DL-based intrusion detection systems. This limitation negatively influences precision in identifying and countering security threats. Specifically, current IDSs often struggle to adapt in real time to the evolving nature of cyber threats, which impairs their efficacy in dynamic network environments. This research aims to address these issues by exploring and devising solutions within DL frameworks for real-time cyber threat analysis. The primary issue centers on inadequate feature learning and robust dataset management, critically affecting the accuracy and responsiveness of IDSs amidst rapidly evolving cyber threats.

3.Methodology

Figure 1 illustrates the functional composition of the CNN-based network IDS model, encompassing three modules: “data preprocessing, feature self-learning, and classification.” The model employs CNN training on a “preprocessed original sample dataset” to ensure effective convergence, which is further enhanced through iterative “feature extraction and refinement.”

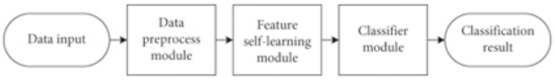

Fig. 1. Intrusion detection architecture for networks using CNN

3.1 CNNs

Network intrusion detection methods utilizing convolutional neural networks significantly surpass traditional ML techniques in classification accuracy. Recent years have witnessed an increasing integration of CNNs into network IDS, owing to their superior feature learning capabilities and semi-supervised nature. This allows them to abstractly represent both low-level and high-level features in intrusion traffic data.

Figure 2 presents a CNN—a type of neural network that replaces regular matrix multiplication operations with convolution operations in one or more layers [26]. Convolution, typically used in image identification tasks, is a specific type of linear operation. Each convolution in the network corresponds to a different aspect of the image, and the low-level convolution of the network continually learns fundamental image attributes such as color, edge, space, and frequency [27].

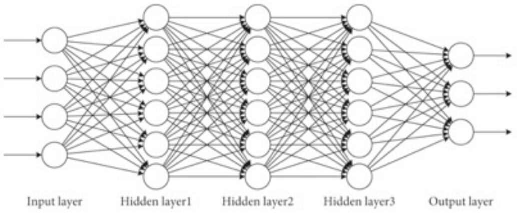

Fig. 2. CNN

The proposed CNN ensures precise classification and effectively addresses the challenge of parameter explosion in neural networks. Three key concepts underpin the CNN: pooling, parameter sharing, and local perception. Because of local perception, individual neurons in the hidden layer need only connect to small portions of the input pixels rather than the entire array. This local perception is achieved through convolutional computations performed by the convolutional layers on incoming data, facilitated by the convolution nucleus in CNNs.

To optimize CNN-based network intrusion detection, a refined methodology was introduced. The architectural design of the model includes adjustments in depth and width, the integration of skip connections for effective gradient flow, and the implementation of attention mechanisms to concentrate on relevant features. Enhanced data augmentation techniques, including both temporal and spatial variations, along with normalization methods, are employed to improve the robustness and generalizability of the model. Training strategies incorporate dynamic learning rate schedulers, the application of transfer learning from pre-trained models, and the use of ensemble techniques to elevate overall performance. Regularization methods, such as dropout layers, are utilized to prevent overfitting. This comprehensive strategy aims to improve the classification accuracy as well as address challenges like parameter explosions, ensuring that the CNN efficiently learns and abstractly represents features in intrusion traffic data.

3.2 Data Pre-processing

For a CNN to effectively process accurate intrusion data, which is typically in 1-dimensional vector form, it must first be transformed into 2-dimensional data akin to an image. This transformation is facilitated by the “data preprocessing module,” which characterizes data by converting text features into numerical values and standardizing numerical features. To ensure that all valuable information from the initial data sample is preserved, a data-driven transformation technique is utilized. This technique augments the sample by incorporating additional features and maintaining all information from the original sample. Standard data are used to populate these expanded characteristics, thereby preserving all valuable information from the initial data sample. The enhanced attributes serve to increase the information capacity of the data sample, enhance the differentiation between data categories in the sample space, and consequently improve detection accuracy to a certain extent.

Given that the characteristic values of intrusion detection data vary widely, processing this data during data preprocessing is crucial. In this work, the widely employed z-score standardization method is used to standardize the numeric properties of intrusion data, as illustrated in the following formula:

$$x_i^{'} = \frac{x_i - \overline{x}}{v} \tag{1}$$

In the formula

$$\overline{x} = (1/n)\sum_{i-1}^{n} x_i, v = \sqrt{(1/n-1)\sum_{i=1}^{n} (x_i - \overline{x})^2}$$

x_i following normalization is the "characteristic value of the dimension" that corresponds to the sample data, represents the pre-standardization characteristic value, and n indicates the total sample numbers.

3.3 Feature Self Learning

The feature self-learning module primarily utilizes CNNs to map, learn, and generate new features from the initial input sample, as well as to autonomously extract and learn valuable features from those samples. Eslami et al. [28] methodically elucidated convolutional neural networks. Figure 3 displays the architecture of the feature self-learning module developed in this paper, which is founded on CNN principles. The key components of the DL method used by the "feature self-learning module" include activation functions, convolution and pooling operations, dropout, and the ADAM optimization algorithm [29].

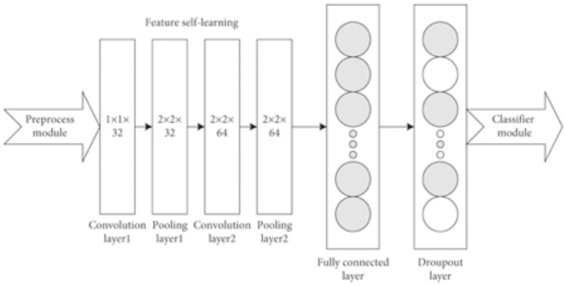

Fig. 3. Feature self-learning module

CNNs present an effective solution to the issue of neural network parameter explosion. They rely on three fundamental concepts: pooling, local perception, and parameter sharing. The convolution procedures enable CNNs to achieve local perception. Equation (2) details the convolution process, where s = feature map output data, x = input sample data, is the kernel function's weight, b = offset value, and f = activation function.

$$s = f(x \times w + b) \tag{2}$$

To further minimize the parameters of the neural network, CNNs employ parameter sharing. Essentially, this indicates that all hidden neurons in the network share the same bias parameters and weight set, leading to the assumption that statistical characteristics based on different image regions are typically identical [30]. A feature map is produced by a set of weight bias parameters and a weight set, but its representation capability is limited. Therefore, in practical applications, a convolutional layer generates multiple feature maps. Primarily, feature pooling aims to render the features more concise.

In CNNs, pooling operations typically

involve identifying either the greatest or the average value among several features in a surrounding region. As such, average pooling and maximal pooling are two common types of pooling operations in CNNs. This study adopts a model that utilizes the average pooling procedure. The tanh function, also known as the double-curving function, is one of the most frequently used activation functions in CNNs [31]. The tanh function is particularly effective when features are highly divergent as it amplifies the effect of features across the cycle. Consequently, the tanh function is employed as the activation function in the CNN of this study.

To address the issue of overfitting, common strategies include regularization, early stopping, sample size augmentation, dropout implementation, and batch normalization. In this study, overfitting is mitigated by introducing a dropout layer between the classifier and the feature self-learning module. The dropout method involves training the model by randomly omitting neurons from the neural network according to probability p. During testing, all neurons are active, which helps prevent overfitting through feature synergy [32]. Each subnetwork, trained using the dropout method, represents a version of the original neural network. This approach enables the generation of 2n models for every hidden node n in our neural networks. To enhance the capability and generalizability of the model, the sub-model prediction outcomes were averaged while the predictions were generated. According to Choraś et al. [33], the richest network structure is created at p = 0.05, wherein dropout has the most effective effect.

Regarding first-order optimization algorithms for deep learning, the ADAM algorithm has gained prominence. As noted by Kingma and Ba [31], the ADAM method combines the advantageous features of adaptive gradient and root mean square propagation algorithms. It applies multiple adaptive learning rates for each parameter, facilitating faster convergence. Network intrusion detection systems often grapple with issues of data sparsity and noise. Therefore, this paper opts for the ADAM method as the optimization algorithm for the CNN model.

3.4 Classifier Module

After the self-learning module has identified specific traits, the classifier utilizes them to generate the final test outcomes. This section describes the CNN training using the “Softmax classifier.”

The Softmax classifier is illustrated in equation (3). $x^{(i)}$ denotes the i-th data sample and j represents the j-th weight vector,

$$y_j = \frac{e^{\theta_j} x^{(i)}}{\sum_k \theta_k x^{(i)}} \tag{3}$$

Mean square error (MSE) and cross entropy error (CEE) are loss functions employed in different contexts. Linear regression models, which focus on regression problems, predominantly use the “mean square error loss function” due to its effectiveness in value predictions. In contrast, the “cross-entropy error loss function” is commonly used in logistic regression for prediction probability, or classification problems. Convolutional neural network models typically employ the “cross-entropy error loss function” for their operations.

4. Result and Analysis

4.1 Experiment and Setting

The computational setup for our experiment includes an Intel Core i7-3920XM processor, a 1 TB solid-state drive (SSD), and 32 GB of RAM. The experiments are conducted in a Docker 19.03.5 container virtualization environment running on Ubuntu 16.04, with Python 3.7 and TensorFlow 1.12.0 utilized for deep learning implementation.

This study involves performing a series of network intrusion classification tests, each comprising a normal (negative) sample and several attack (positive) samples in each dataset. The number of labeled classes varies across datasets. Consequently, a matrix that consolidates multiple classes is constructed to demonstrate the effectiveness of the model when applied to specific datasets. The confusion matrix includes data pertaining to both observed and expected classes. It provides four key outcomes: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

Regarding multiclass classification tasks, these four outcomes did not match the two approaches. First, a typical sample may be accurately predicted by TN. In equation (4), we may find FP, where i denotes the number of samples misclassified FP_i as belonging to the normal attack class. N denotes the number of attack classes. TP_i indicates the exact predictor of the i-th attack category, and TP denotes the sum of assault samples that are tagged as their respective category using equation (5). FN denotes the total number of attack samples classified in the incorrect normal class. Equation (6) can be used to calculate FN_i, i.e., the number of attack class samples incorrectly classified as normal:

$$FP = \sum_{i=1}^{N} FP_i \tag{4}$$

$$TP = \sum_{i=1}^{N} TP_i \tag{5}$$

$$FN = \sum_{i=1}^{N} FN_i \tag{6}$$

To evaluate the performance of the model on the dataset, these four outcomes are used to derive five assessment metrics, with mathematical adjustments made to align with the multiclass NIDS definition and the system’s terminology. The accompanying equation defines the assessment indicator.

- The FAR reveals that for every assault category, the test set with a normal sample is incorrectly labeled as normal sample rates, also described by the term "false positive rate" (FPR).

$$FAR = \frac{FP}{FP + TN} \tag{7}$$

- F-Score denotes the harmonic mean of the precision (P) & recall (R) indicators

$$F - score = \frac{2 \times \Pr ecision \times \operatorname{Re} call}{\Pr ecision + \operatorname{Re} call} \tag{8}$$

- The recall of the classifier is its accuracy; test set attack samples should exhibit the same labeled attack rate; it is also called detection rate (DR), sensitivity, or true positive rate (TPR):

$$\mathrm{Re}\,call = \frac{TP}{TP + FN} \tag{9}$$

- Precision refers to the accuracy of the classifier or the percentage of test samples accurately identified as attacks.

$$\mathrm{Pr}\,ecision = \frac{TP}{TP + FP} \tag{10}$$

- Accuracy metric reveals the actual rate of prediction across the test set

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

4.2 Experimental Dataset

This study introduces a network IDS model and evaluates it using the NSL-KDD dataset, a widely recognized benchmark in security-related research, due to its status as the de facto standard for network intrusion detection [35].

4.2.1 NSL-KDD Dataset

The NSL-KDD dataset allows comprehensive experimentation, as both its training and test sets comprise a substantial number of records. Despite its limitations in perfectly representing real-world networks, it remains a cornerstone in network intrusion detection research. The NSL-KDD dataset, available online [36], comprises records characterizing network connections based on 41 attributes. Each record in the dataset is classified as either a normal or an attack, with 39 different attack types categorized into four main groups. The training and test sets feature 22 common attacks, with an additional 17 unique attacks present only in the test set. Table 2 details the specifics of the NSL-KDD dataset.

Table 2. NSL-KDD data details

		Data types	
		Training sets	Test sets
Normal		67343	9711
Attacks	U2R	52	67
	R2L	995	2887
	Probe	11656	2421
	DoS	45927	7458
Total		125973	22544

4.3 Experiment Result

4.3.1 Experiment on NSL-KDD

Over 30 iterative training sessions, the NSL-KDD dataset was processed. The test results are compiled in Table 3. The first four columns present the arithmetic mean of the experimental outcomes from the pretraining (w/) and non-pretraining (w/o) phases, as referenced in [37]. The fifth column shows the results from reference [43].

Figure 5 facilitates visual comparisons of the evaluation indices across all models. The experimental data in Table 3 attest to the high efficacy of the proposed network intrusion detection model, achieving an accuracy rate of 97.09% and a false alarm rate of 0.87%. The detection results indicate a reduced false alarm rate of 1.74% and an enhanced accuracy rate of 92.66% compared to the most optimal detection results documented in the literature [37, 28]. Figure 4(b) graphically displays the detection rates of NSL-KDD for each attack type. Refer to Figure 4(b) for a depiction of how an increased training set size for specific attack types correlates with higher detection rates. Additionally, the experiment confirms that attack-type data not included in the training set can still be accurately classified.

Table 3. Experiment results on NSL-KDD dataset

Metrics	CNID (this paper) (%)	ICNN [38] %	DBN [37] (%)	GRU-RNN [37https://www.hindawi.com/journals/ddns/2020/4705982/ - B40] (%)	LSTM-RNN [37] (%)	DNN [37] (%)
FAR	0.88	2.32%	1.74	2.58	2.03	2.75
F1-Score	98.50	—	93.33	89.79	90.99	86.05
Recall	97.15	—	89.56	82.95	83.70	77.19
Precision	99.99	93.65%	97.43	97.02	97.52	96.73
Accuracy	97.10	91.79%	92.66	89.58	90.39	85.74

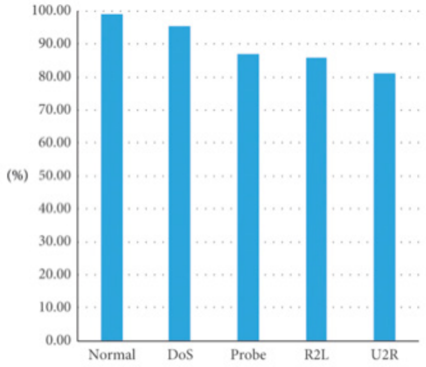

Fig. 4. NSL-KDD dataset for each class with the detection rate

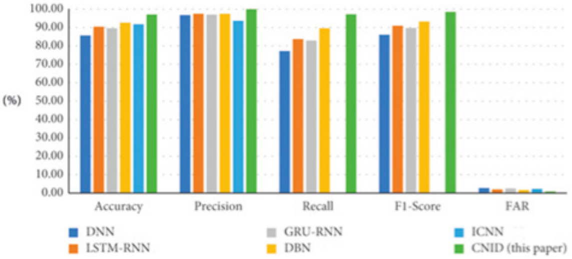

Fig. 5. Experiment results for the NSL-KDD dataset

4.3.2 Comparison with Other Related Works

The precise calibration of the model's assessment metrics is crucial in determining its effectiveness in detecting network intrusions. A lower FAR indicates the classifier's proficiency, especially when metrics such as accuracy, recall, F-Score, and precision are high. In an ideal classifier, the FAR is zero, with recall and precision equal. The empirical results obtained from the KDD-CUP 99 and NSL-KDD datasets are compared with the four deep learning models discussed in recent studies [37, 38]. Our proposed model outperforms those in the literature, achieving a higher accuracy rate of 97.10% on the NSL-KDD dataset compared to the literature's rate of 92.67% [37]. A study in the literature introduced a novel network intrusion detection model [39] aimed at addressing data imbalance issues.

This model leverages CNNs to autonomously extract traffic features from the initial dataset and adjusts the weight coefficient of the cost function based on category counts. When evaluated on the NSL-KDD dataset for large-scale network intrusion detection, this model exhibits performance that is inferior to the one proposed in our study. Our model also demonstrates enhanced performance compared to the previous model when tested on the KDD-CUP99 dataset. The earlier model used the CNN model Lenet-5 and applied one-hot encoding (OHE) and normalization techniques for feature matrix handling. Previous literature [40, 41] has employed CNNs for intrusion detection across various domains [42, 43]. Although there is room for improvement in distinguishing between various types of attacks, our proposed approach shows superior generalization ability, effective detection of unknown attack types, and notable proficiency in distinguishing between normal and attack data.

4.3.3 Deployment Challenges in Real-world Cybersecurity Environments

Implementing the proposed CNN-based framework in actual cybersecurity settings entails numerous challenges and considerations that require meticulous planning. The integration of this framework into pre-existing cybersecurity infrastructures is a significant concern, demanding seamless compatibility with various systems, protocols, and security architectures. Addressing these compatibility issues is essential for smooth integration. Another critical aspect is resource allocation, including the requirements for computational power, memory, and processing capabilities, which are vital for the efficient operation of the framework.

Furthermore, the model's ability to adapt to constantly evolving cyber threats is a crucial factor. Given the dynamic nature of cybersecurity, where threats continually change, the framework must rapidly adapt to new attack patterns and scales to accommodate an increasing variety of threats over time. Practical implementation challenges, such as ensuring data privacy, adhering to legal compliance, and maintaining continuous monitoring and updates, are crucial for ensuring the success and legitimacy of the model in real-world applications. These considerations highlight the need for a comprehensive and strategic approach to deploying the proposed CNN-based framework effectively in practical cybersecurity environments.

4.3.4 Interpreting CNN-based Models: Building Trust and Understanding Decision Making

An inherent challenge in deploying CNN-based models for network intrusion detection is interpreting the complex decision-making processes inherent in these models. The opaque, "black box" nature of CNN complicates efforts to fully understand the rationale behind specific decisions, leading to concerns about transparency and interpretability in a field where trust in decision-making is crucial. Overcoming this interpretability barrier is essential for gaining acceptance and facilitating the adoption of the CNN-based framework. Therefore, developing methods to clarify the decision logic of the model, such as employing layer-wise relevance propagation or attention mechanisms, is imperative. By enabling cybersecurity professionals to interpret and validate the decisions of the model, trust is built, transparency is enhanced, and effective collaboration between human analysts and automated systems is fostered. Addressing the interpretability of CNN-based models is vital for building trust and reinforcing the reliability and acceptance of the framework in real-world cybersecurity deployments.

4.3.5 Scalability Considerations in Large-scale Network Environments

Scalability is a crucial aspect for the deployment of the proposed CNN-based framework, especially in large-scale network environments, and it requires a thorough exploration of potential scalability challenges. With the expansion of network sizes, the computational demands on the framework may increase significantly. This escalation presents challenges in terms of processing power, memory requirements, and the capacity to manage an increasing volume of network traffic efficiently. To ensure scalability, it is essential to identify and address potential bottlenecks while optimizing algorithms for enhanced efficiency. Furthermore, the framework must accommodate real-time processing and sustain high performance, even with large datasets. Implementing robust strategies for distributed computing, parallel processing, or cloud-based solutions could be crucial in overcoming these scalability challenges. A comprehensive analysis of scalability is imperative to ensure that the proposed CNN-based framework remains effective and performs optimally in the face of the complexities associated with large-scale network environments.

4.3.6 Assessing Computational Efficiency

The assessment of computational efficiency is a critical factor in determining the practical applicability of the proposed CNN-based framework for network intrusion detection. Although the significance of assessing computational efficiency is acknowledged, the discussion lacks specific details on the exact measures or comparative benchmarks used, leading to ambiguity regarding their practical impact. A detailed examination should consider metrics, such as processing time, memory utilization, and throughput, with explicit numerical values or performance benchmarks for clarity. Including a table that summarizes these metrics would significantly enhance the transparency of the assessment, enabling readers to grasp the trade-offs and advantages of implementing the framework in real-world scenarios. Incorporating precise information on computational efficiency metrics and their implications fortifies the assessment and provides invaluable insights to cybersecurity experts and researchers evaluating the proposed model.

Table 4. Computational efficiency of the proposed CNN framework

Metric	Proposed Framework in NSL-KDD Dataset	Comparison Model in KDD Cup Dataset
Processing Time (ms)	15	20
Memory Utilization (%)	80	90
Throughput (Mbps)	120	100

Table 4 enhances the understanding and practical significance of the proposed CNN-based framework by offering a detailed evaluation of its computational efficiency. This comparison is made between the proposed framework applied to the NSL-KDD dataset and a benchmark model used on the KDD-CUP dataset, focusing on metrics that are key indicators of the framework's utility in network intrusion detection.

Processing Time (ms): The time required for the proposed framework to perform intrusion detection tasks on the NSL-KDD dataset is significantly shorter, recorded at 15 ms, compared to the 20 ms required by the model on the KDD-CUP dataset. This metric highlights the efficiency of the framework in rapidly processing and analyzing network data for intrusion detection purposes.

Memory Utilization (%): On the NSL-KDD dataset, the proposed framework exhibited a lower memory utilization of 80%, in contrast to the comparison model, which uses 90% of available memory on the KDD-CUP dataset. Reduced memory utilization indicates a more resource-efficient operation.

Throughput (Mbps): For the proposed framework, the throughput—the rate at which tasks are processed—is higher at 120 Megabits per second on the NSL-KDD dataset. This is compared to 100 megabits per second achieved by the model on the KDD-CUP dataset. A greater throughput reflects a swifter and more efficient processing capacity.

Table 4 provides quantitative insight into the computational efficiency of the proposed CNN-based framework, facilitating an evaluation of its applicability and effectiveness in real-world network intrusion detection scenarios. The combination of shorter processing times, reduced memory utilization, and increased throughput points to a more proficient and effective performance of the proposed framework on the evaluated dataset.

4.4 Limitations

1. Lack of Explainability: CNNs are often criticized for their "black-box" nature, which obscures the interpretability of their decision-making processes in cyber threat detection
2. Resource Intensity: CNNs can demand significant computational resources, which may pose challenges for real-time deployment, particularly in environments with limited resources.
3. Ethical consideration: Ensuring informed participant consent and robust data privacy protection is crucial, safeguarding the rights and welfare of individuals involved in the research.

5. Conclusions

In the field of network security, intrusion detection is critical. Despite the abundance of information on NITK, recent research has paid limited attention to multiclass NITK. This study introduces and refines a CNN for multiclass network IDS. The system requirements for the experiment included 32 GB RAM, a 1 TB SSD, Ubuntu 16.04 OS, and a Docker 19.03.5 container virtualization environment. The experiment compared outcomes using various DL models, including DNN, LSTM-RNN, GRU-RNN, DBN, KNN, and ICNN, against the NSL-KDD datasets. The testing results demonstrate that the proposed network IDS model outperforms contemporary techniques in terms of accuracy, recall, reduced false positive rate, and identification of unknown threats.

Future research should aim at enhancing the accuracy of the model proposed in this study for multiclass testing and should explore methods to improve the classification of various types of unknown threats. The experiment in this paper utilized a dataset that underwent human optimization and processing. Subsequent studies will focus on a new network IDS dataset that is currently in development. This forthcoming dataset will extract relevant information from actual network traffic characteristics to further validate the approach proposed in this research.

References

[1] Wu, Jiangxing. Cyberspace mimics defense. Cham: Springer International Publishing, 2020.

[2] Dhoni, Pan Singh, and Ravinder Kumar. "Synergizing Generative AI and Cybersecurity: Roles of Generative AI Entities, Companies, Agencies, and Government in Enhancing Cybersecurity." (2023).

[3] Samtani, Sagar, Maggie Abate, Victor Benjamin, and Weifeng Li. "Cybersecurity as an industry: A cyber threat intelligence perspective." The Palgrave Handbook of International Cybercrime and Cyberdeviance (2020): 135-154.

[4] Zhao, Jingyuan, Xuebing Han, Minggao Ouyang, and Andrew F. Burke. "Specialized deep neural networks for battery health prognostics: Opportunities and challenges." Journal of Energy Chemistry (2023).

[5] de Assis, Marcos VO, Luiz F. Carvalho, Joel JPC Rodrigues, Jaime Lloret, and Mario L. Proença Jr. "Near real-time security system applied to SDN environments in IoT networks using convolutional neural network." Computers & Electrical Engineering 86 (2020): 106738.

[6] Alzubaidi, Laith, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie, and Laith Farhan. "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions." Journal of Big Data 8 (2021): 1-74.

[7] Aziz, Layla Abdel-Rahman, and Yuli Andriansyah. "The Role Artificial Intelligence in Modern Banking: An Exploration of AI-Driven Approaches for Enhanced Fraud Prevention, Risk Management, and Regulatory Compliance." Reviews of Contemporary Business Analytics 6, no. 1 (2023): 110-132.

[8] Kaur, Ramanpreet, Dušan Gabrijelčič, and Tomaž Klobučar. "Artificial intelligence for cybersecurity: Literature review and future research directions." Information Fusion (2023): 101804.

[9] Hong, Rui-Fong, Shih-Cheng Horng, and Shieh-Shing Lin. "Machine learning in cyber security analytics using NSL-KDD dataset." In 2021 International Conference on Technologies and Applications of Artificial Intelligence (TAAI), pp. 260-265. IEEE, 2021.

[10] Gupta, Rajesh, Sudeep Tanwar, Sudhanshu Tyagi, and Neeraj Kumar. "Machine learning models for secure data analytics: A taxonomy and threat model." Computer Communications 153 (2020): 406-440.

[11] Krichen, Moez. "Convolutional neural networks: A survey." Computers 12, no. 8 (2023): 151.

[12] AlHaddad, Ulaa, Abdullah Basuhail, Maher Khemakhem, Fathy Elbouraey Eassa, and Kamal Jambi. "Ensemble Model Based on Hybrid Deep Learning for Intrusion Detection in Smart Grid Networks." Sensors 23, no. 17 (2023): 7464.

[13] Dong, Tian, Song Li, Han Qiu, and Jialiang Lu. "An interpretable federated learning-based network intrusion detection framework." arXiv preprint arXiv:2201.03134 (2022).

[14] Moustafa, Nour, Nickolaos Koroniotis, Marwa Keshk, Albert Y. Zomaya, and Zahir Tari. "Explainable Intrusion Detection for Cyber Defences in the Internet of Things: Opportunities and Solutions." IEEE Communications Surveys & Tutorials (2023).

[15] Nassar, Ahmed, and Mostafa Kamal. "Machine Learning and Big Data Analytics for Cybersecurity Threat Detection: A Holistic Review of Techniques and Case Studies." Journal of Artificial Intelligence and Machine Learning in Management 5, no. 1 (2021): 51-63.

[16] Otoum, Yazan, Dandan Liu, and Amiya Nayak. "DL-IDS: a deep learning–based intrusion detection framework for securing IoT." Transactions on Emerging Telecommunications Technologies 33, no. 3 (2022): e3803.

[17] Jothi, B., and M. Pushpalatha. "WILS-TRS—A novel optimized deep learning based intrusion detection framework for IoT networks." Personal and Ubiquitous Computing 27, no. 3 (2023): 1285-1301.

[18] [18] Ferrag, Mohamed Amine, Othmane Friha, Leandros Maglaras, Helge Janicke, and Lei Shu. "Federated deep learning for cyber security in the internet of things: Concepts, applications, and experimental analysis." IEEE Access 9 (2021): 138509-138542.

[19] Sriram, S., R. Vinayakumar, Mamoun Alazab, and K. P. Soman. "Network flow based IoT botnet attack detection using deep learning." In IEEE INFOCOM 2020-IEEE conference on computer communications workshops (INFOCOM WKSHPS), pp. 189-194. IEEE, 2020.

[20] Aslan, Ömer, and Abdullah Asim Yilmaz. "A new malware classification framework based on deep learning algorithms." Ieee Access 9 (2021): 87936-87951.

[21] Lansky, Jan, Saqib Ali, Mokhtar Mohammadi, Mohammed Kamal Majeed, Sarkhel H. Taher Karim, Shima Rashidi, Mehdi Hosseinzadeh, and Amir Masoud Rahmani. "Deep learning-based intrusion detection systems: a systematic review." IEEE Access 9 (2021): 101574-101599.

[22] Amanullah, Mohamed Ahzam, Riyaz Ahamed Ariyaluran Habeeb, Fariza Hanum Nasaruddin, Abdullah Gani, Ejaz Ahmed, Abdul Salam Mohamed Nainar, Nazihah Md Akim, and Muhammad Imran. "Deep learning and big data technologies for IoT security." Computer Communications 151 (2020): 495-517.

[23] aveed, Danish, Tianhan Gao, and Muhammad Taimoor Khan. "SDN-enabled hybrid DL-driven framework for the detection of emerging cyber threats in IoT." Electronics 10, no. 8 (2021): 918.

[24] Lee, In. "Cybersecurity: Risk management framework and investment cost analysis." Business Horizons 64, no. 5 (2021): 659-671.

[25] Venkatraman, Sitalakshmi, Mamoun Alazab, and R. Vinayakumar. "A hybrid deep learning image-based analysis for effective malware detection." Journal of Information Security and Applications 47 (2019): 377-389.

[26] Ketkar, Nikhil, Jojo Moolayil, Nikhil Ketkar, and Jojo Moolayil. "Convolutional neural networks." Deep Learning with Python: Learn Best Practices of Deep Learning Models with PyTorch (2021): 197-242.

[27] Zhong, Jiachen, Junying Chen, and Ajmal Mian. "DualConv: Dual convolutional kernels for lightweight deep neural networks." IEEE Transactions on Neural Networks and Learning Systems (2022).

[28] Eslami, Elham, and Hae-Bum Yun. "Attention-based multi-scale convolutional neural network (A+ MCNN) for multi-class classification in road images." Sensors 21, no. 15 (2021): 5137.

[29] D. Kingma and J. Ba, "ADAM: a method for stochastic optimization," 2017, https://arxiv.org/abs/1412.6980v9.

[30] Wang, Zhendong, Yong Zeng, Yaodi Liu, and Dahai Li. "Deep belief network integrating improved kernel-based extreme learning machine for network intrusion detection." IEEE Access 9 (2021): 16062-16091.

[31] Guo, Yanhua, Lei Sun, Zhihong Zhang, and Hong He. "Algorithm research on improving activation function of convolutional neural networks." In 2019 Chinese Control And Decision Conference (CCDC), pp. 3582-3586. IEEE, 2019.

[32] Kan, Xiu, Yixuan Fan, Zhijun Fang, Le Cao, Neal N. Xiong, Dan Yang, and Xuan Li. "A novel IoT network intrusion detection approach based on adaptive particle swarm optimization convolutional neural network." Information Sciences 568 (2021): 147-162.

[33] Choraś, Michał, and Marek Pawlicki. "Intrusion detection approach based on optimized artificial neural network." Neurocomputing 452 (2021): 705-715.

[34] [34] Li, Yanmiao, Yingying Xu, Zhi Liu, Haixia Hou, Yushuo Zheng, Yang Xin, Yuefeng Zhao, and Lizhen Cui. "Robust detection for network intrusion of industrial IoT based on multi-CNN fusion." Measurement 154 (2020): 107450.

[35] Su, Tongtong, Huazhi Sun, Jinqi

Zhu, Sheng Wang, and Yabo Li. "BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset." IEEE Access 8 (2020): 29575-29585.

[36] NSL-KDD dataset, https://github.com/defcom17/NSL_KDD.

[37] W. Elmasry, A. Akbulut, and A. H. Zaim, “Empirical study on multiclass classification-based network intrusion detection,” Computational Intelligence, vol. 35, no. 4, pp. 915–954, 2019.

[38] H. Yang and F. Wang, “Network intrusion detection model based on improved convolutional neural network,” Journal of Computer Applications, vol. 39, no. 9, pp. 2604–2610, 2019.

[39] K. Wu, Z. Chen, and W. Li, “A novel intrusion detection model for a massive network using convolutional neural networks,” IEEE Access, vol. 6, pp. 50850–50859, 2018.

[40] [40] P. Liu, “An intrusion detection system based on convolutional neural network,” in Proceedings of the 2019 11th International Conference on Computer and Automation Engineering ICCAE, pp. 62–67, Perth, Australia, February 2019.

[41] H. Yang and F. Wang, “Wireless network intrusion detection based on improved convolutional neural network,” IEEE Access, vol. 7, pp. 64366–64374, 2019.

[42] H. M. Song, J. Woo, and H. K. Kim, “In-vehicle network intrusion detection using deep convolutional neural network,” Vehicular Communications, vol. 21, Article ID 100198, 2020.

[43] Y. Xiao, C. Xing, T. Zhang, and Z. Zhao, “An intrusion detection model based on feature reduction and convolutional neural networks,” IEEE Access, vol. 7, pp. 42210–42219, 2019.

[44] Haq, Mohd Anul, Mohd Abdul Rahim Khan, and Talal AL-Harbi. "Development of PCCNN-Based Network Intrusion Detection System for EDGE Computing." Computers, Materials & Continua 71, no. 1 (2022).