Journal of Theoretical and Applied Information Technology
30st June 2026. Vol.104. No.12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ENHANCING NETWORK SECURITY WITH CONVOLUTIONAL NEURAL NETWORKS: AN INTRUSION DETECTION MODEL USING THE NSL-KDD DATASET

AMMAR D. JASIM[1], SAIF S. KAREEM[2]

[1]Department of Computer Networks Engineering, College of Information Engineering, Al-Nahrain University, Jadriya, Baghdad, Iraq
[2]Al-Nahrain University, Baghdad, Iraq

[1]Ammar.alaythawy@nahrainuniv.edu.iq, [2]Saif.salah.k@nahrainuniv.edu.iq

*Saif.salah.k@nahrainuniv.edu.iq

ABSTRACT

Intrusion Detection Systems (IDS) are very significant in the protection of network infrastructure systems against unauthorized malicious attacks and new threats. In the given work, the Deep Neural Network (DNN)-based IDS is suggested to be based on Convolutional Neural Networks (CNNs), and this suggestion is tested on the NSL- Knowledge Discovery in Databases (KDD) benchmark dataset. The proposed model, in contrast to the traditional machine learning techniques using manual feature engineering, provides automatic feature extraction of a hierarchical structure which is based on the time-frequency representation of the network traffic, facilitating detection. To solve the problem of class imbalance, weighted loss functions are used, and categories are represented by categorical encoding, and feature standardization is used to learns the models. The model had an outstanding performance of 99.87% accuracy and AUC 0.9265 and which was quite higher compared to the baseline 5-layer Auto encoder and other traditional methods. The experimental findings indicate the validity of the model because it can differentiate normal and attack traffic with very low false positives and false negatives. As a regularization method, a regular way to do early stopping is to avoid overfitting and to generalize. Although the proposed system proves to be very effective on benchmark data, a few issues are still related to high computational requirements and real-time usability. Future work should focus on optimizing the model for deployment, improving zero-day attack resilience and ensuring robust testing of the model performance under variable and realistic network conditions. This collection of results adds to the combinability of CNN-based models and the most advanced IDS systems for more proactive and intelligent cyber security.

Keywords: *Intrusion Detection Systems, Convolutional Neural Network, Deep Neural Network*.

1. INTRODUCTION

By 2030 over 500 billion devices will be connected to the Internet: a level of digital interconnectivity never seen before [1]. However, this hyper-connectivity holds a phenomenal potential for innovation and operational efficiency across industries; it unquestionably also increases the exposure to cyber threat vectors significantly. The frequency and sophistication cyberattacks have increased in recent years, highlighting the need for more robust and intelligent networks security systems [2, 3]. In order to respond to modern cybersecurity threats, we need not just knowledge of the changing patterns of attack, but also innovative and resilient defenses. Recent developments in data science and artificial intelligence (AI) have provided exciting new pathways for addressing very challenging security problems [4]. In particular, AI-driven techniques have demonstrated high potential to improve network anomaly detection and minimize threat response times [5].A wide range of AI-based intrusion detection techniques have emerged in recent years, demonstrating that the integration of AI with cybersecurity can substantially strengthen network defenses strategies [6].

In order to prevent leakage of confidentiality, integrity, and availability of the digital assets, organizations normally install various security features, including firewalls, antivirus programs, anti-malware programs, and encryption guidelines. The Intrusion Detection System (IDS) is one of such layers of protection, but one of the most important ones. An IDS enables the detection of malicious activity in the network or system by watching the network or system activity and raises an alarm to the system administrators in real time, either directly or via a Security Information and Event Management (SIEM) system [7].

IDSs can be broadly divided into network-based Intrusion Detection (NIDS) and Host-based Intrusion Detection Systems (HIDS) [8]. Components on which NIDS use to monitor the network include network interface cards and a combination of sensors working in promiscuous mode to analyze packets on various endpoints [9]. In case signatures of suspicious behavior are discovered, as they are compared to known signatures of threats, the system will trigger an alert. Conversely, HIDS run on host machines where they monitor the internal processes, system records and changes of file probing signs of attacks [10].

Intrusion Detection system methods may additionally be divided by their detection methodology: signature-based (SIDS), anomaly-based (AIDS), and those that are hybrids of both. Signature-based IDS make use of signatures or pre-determined attack patterns [11]. Such systems are very effective in informed attacks; however, they are ineffective against new or unknown attacks. Anomaly-based IDS, on the other hand, depend on statistical modeling or machine learning to detect when behavior is out of the ordinary and, therefore, is better placed to detect emerging threats [12]. They do note, though, that such systems need a lot of training on quality data in order to accurately flag benign and malicious behavior [13].

There is an urgent demand for advanced intrusion detection. By the year of 2021, it has been projected that the overall cost of cybercrimes will amount to 6 trillion, which demonstrates the drastic impacts that poor cybersecurity can have on economic and technological aspects [14]. Though the conventional tools like firewalls and antivirus programs are critical, they are, in most cases not adequate in terms of combating the newer and more complicated attacks. IDSs offer important forensics data in the middle of an attack and post-attack that helps an organization to respond more efficiently and build a protective security approach [15].

Last but not least, an IDS can just be as good as it is to give the algorithms. It is stated that signature-based systems are accurate when it comes to big threats, whereas anomaly-based systems are more flexible and could even manage zero-day attacks. The appropriate training and application of Anomaly-based IDS will help enable organizations to anticipate the effect and establish effective resistance to the emerging threat to information and communications technology (ICT) architecture [16]. In order to improve the accuracy and performance of detection, various optimization algorithms like metaheuristic algorithms have been added to IDS frameworks in increasing numbers. The use of techniques such as Particle Swarm Optimization (PSO) [17, 18], Genetic Algorithms (GA) [19], Black Hole Optimization [20], Grey Wolf Optimizer [21], Harris Hawks Optimization [22], and Sunflower Optimization (SFO) [23] helps choose the best features, suitable hyperparameters of learning models and enhance the rate of convergence. Such algorithms are of paramount importance to the optimization of efficient operation of IDSs, minimization of false positives, and the provision of flexible, scalable defense measures that are appropriate in network environments that are dynamic and complex [24, 25].

Main Contributions of This Study

- We propose a CNN-based intrusion detection framework for binary classification on the NSL-KDD dataset.
- We combine preprocessing steps such as encoding of categorical features and standardisation of numerical features & training with class weights.
- We evaluate the model against a 5-layer autoencoder baseline using accuracy, precision, recall, F1-score, AUC, loss curves, and confusion matrices.
- We discuss practical limitations related to computational cost, generalisation, and real-world deployment

2. LITERATURE REVIEW

The complex and increased instructions of cyber-attacks within the last few years have required the creation of more enhanced IDS, particularly in dynamic network settings like SDNs. The rule-

based or statistical-based traditional IDS solutions have a hard time keeping both detection accuracy and low FP rates as the threats evolve in patterns. Therefore, the application of machine learning (ML) and deep learning (DL) methods as interesting alternatives in intelligent anomaly detection has been considered by scientists. These methods will help IDS to automatically learn and generalize on the network traffic patterns, and this will enhance its flexibility and strength. There has been some extensive literature devoted to carrying out experiments to compare the performance of these models based on ML and DL in intrusion detection by using popular benchmark datasets, e.g., NSL-KDD and KDD K99. Such experiments discuss various methods, including deep neural networks and convolutional neural networks, hybrid and ensemble models, and show that their accuracy and efficiency of detection are increasing. The following papers described below summarizes and analyses the prime works in this direction, outlining the methods, the data utilized, and the performance results of different IDS models suggested in the recent literature.

A study presented in [26] proposed a deep learning-based intrusion detection model leveraging Deep Neural Networks (DNN), which achieved an accuracy of 75.75% using only six fundamental features: duration, protocol type, Source Bytes, Destination Bytes, count, and srv-count extracted from the NSL-KDD dataset. The findings demonstrated that high detection performance can be achieved using a minimal yet strategically selected set of features.

In contrast, the work in [27] and [28] introduced a novel technique by converting raw network traffic into image representations, which were then analyzed using a Convolutional Neural Network (CNN). This model outperformed traditional machine learning approaches, achieving an accuracy of 79.48%, and illustrated the potential of image-based feature extraction in intrusion detection.

A more advanced method, the Hierarchical Combining of Predictions of a Tree of Classifiers (HCPTC-IDS), was introduced in [29]. This approach integrated several classifiers, Naive Bayes (NB), Fuzzy Logic (FL), RIPPER, Decision Tree (DT), Artificial Neural Network (ANN), and Support Vector Machine (SVM), to achieve a high accuracy of 89.75% while maintaining fast processing speeds (373 microseconds per record) on the NSL-KDD dataset.

An ensemble learning model combining Random Forest, Decision Tree, and DNN was proposed in [30], achieving an overall accuracy of 85.2%. This adaptive approach reinforced the value of combining diverse models to enhance detection performance.

The study in [31] introduced the Improved Conditional Variational Auto-Encoder Deep Neural Network (ICVAE-DNN), also using NSL-KDD data. This model was benchmarked against six other classifiers, KNN, Multinomial Naive Bayes, Random Forest, SVM, DNN, and Deep Belief Network (DBN), and achieved the highest accuracy of 85.97%, demonstrating superior learning of latent network traffic patterns.

In [32], fuzzy logic was employed for anomaly detection, resulting in an accuracy of 84.54%. Meanwhile, an Artificial Neural Network (ANN) model proposed in [33] achieved 81.2% accuracy on the same dataset. These results further indicate the competitiveness of alternative soft computing techniques in intrusion detection.

Kevric et al. [34] illustrates hybridization as it includes combination of Random Tree and NB Tree classifies. With this method, an accuracy of 89.24% was achieved on the original KDD dataset, which is higher than single-tree Models and better illustrates the advantages of ensemble learning. The study also recognized the potential of autoencoders (AEs) to enhance feature representation through data compression and noise filtering, as shown in several studies such [35, 36, 37, 38].

Lastly, Anisa et al. Zaire et al.[39] compared the classification performance of SVM and Naive Bayes classifiers were in an IDS application. The results manifestly illustrated that SVM outperformed Naive Bayes with an extensive margin, reaching 97% detection rate, and consequently validated SVM's resilience in network security scenarios.

3. DATASET

One of the most popular cyber security research datasets, KDD99, was developed in 1999 [40]. Redundancy and an excessive quantity of records in both the train and test datasets, which make it challenging to deal with the complete dataset in experiments, are some issues that academics have found after years of studying KDD99. In order to address the aforementioned drawbacks, a more recent version, NSL-KDD, was put out in [41]. For cyber security research, NSLKDD has been

regarded as the new benchmark dataset since 2009. The NSL-KDD dataset comprises 22,544 records from KDDTest+ and 125,973 records from KDD-Train+. A dataset called NSL-KDD has been proposed to address a number of intrinsic issues [42] with previous iterations (like KDD-Cup99) used for network intrusion detection. Due to the dearth of publicly available datasets for network-based intrusion detection systems, the dataset is frequently regarded as the most popular recent network intrusion datasets that can be used as an efficient benchmark to compare various intrusion detection techniques, along with UNSW-NB15 and CICIDS-2017, even though it may not be an exact representation of actual networks. 41 features, which fall into four distinct feature categories, basic features, time-based traffic features, connection-based traffic features, and content features, represent each record [43]. Each record has 21 hypothesized label classes, which stand for attack and normal records, respectively. Every record between two hosts on a network is regarded by the cyber security sector as a session, which is a link between two pairs. KDDTrain+'s probability distribution differs from KDDTest+'s. Some of the assaults in the test dataset are not present in the training data. In order to evaluate the classifier's capacity to identify new assaults, the testing dataset includes 14 additional attack types that are not included in the training dataset, whereas the training dataset has 24 distinct attack types. All things considered, NSL-KDD offers a fresh concept that enhances KDD99. For instance, NSL-KDD does not view probing as an attack unless the number of repetitions exceeds a certain threshold, but KDD99 views it as an assault. Figure 1 gives an elaborate scheme of the NSL-KDD dataset, which is a publicly used benchmark in network intrusion detection studies. The regular bar graph is used to show the big difference in the count of records between the training and test subsets, which amounted to 125.973 and 22.544 instances, respectively. It is also noted that the distribution of the type of attacks is asymmetric, with the training set containing 24 different categories, and the test set containing 14 different types, making it an important feature to test the capability of a model to detect previously unseen intrusions. On its part, the inset pie chart supplements this by showing the distribution of the 41 features in the dataset in a well-structured manner concerning four factors: basic, time-based, connection-based, and content features. Collectively, these aspects summarize the structural and functional aspect of the dataset, which further contribute to its strength in being a standard and test benchmark in the determination of the effectiveness of the network-based intrusion detection systems.

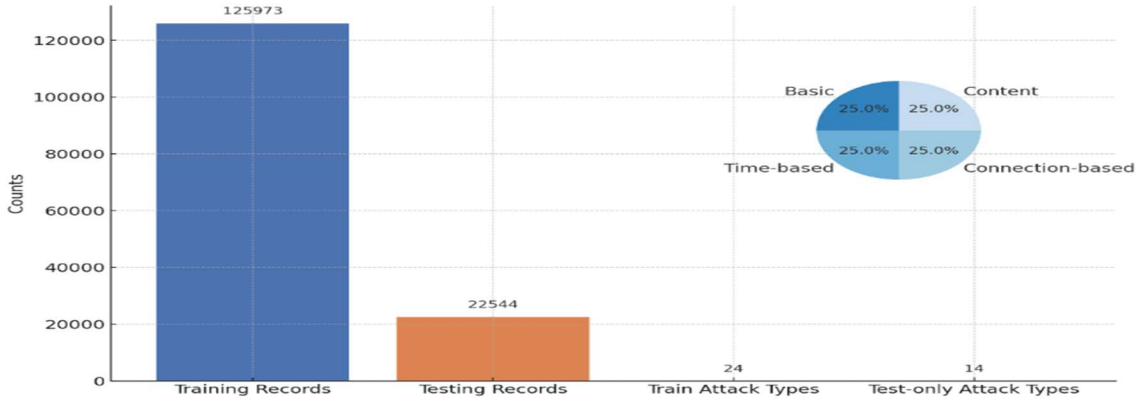

Figure 1. Overview of the NSL-KDD dataset.

4. PROPOSED METHODOLOGY

Network intrusion attempts have the potential to seriously impair a system's security and dependability in an operating setting, making it susceptible to cyberattacks. Different forms of network intrusions may be identified and classified by precise time-frequency analysis of attack patterns in network traffic, which have varying properties. Both temporal and spectral information can be concurrently provided via an appropriate Time-Frequency Representation (TFR). Since the TFR of network traffic data may be saved as pictures, the suggested method for categorizing different types of network incursions is based on machine learning (ML)-based image classification algorithms. In order to train a model to identify

certain attack categories based on pixel or vector patterns in an image, the intrusion detection job is restated as an image classification issue. For classification, conventional machine learning methods like support vector machines (SVM), decision trees, and random forests can be applied. However, in order to use these models, the most pertinent features from the photos must be manually extracted by domain experts. The more infiltration categories there are, the more complicated this procedure gets. Deep learning algorithms, especially Convolutional Neural Networks (CNNs), automatically extract the most important and instructive characteristics for every class in order to overcome these difficulties. CNN-based models have shown state-of-the-art performance in computer vision applications and have been widely employed in picture categorization. CNN is used as the main intrusion classification model in this investigation. However, noise may mask attack characteristics in low-impact or mild incursions, making it challenging for ML models to successfully identify trends in TFR pictures [44]. To get over this restriction, more pertinent characteristics are included in the frequency and temporal domains to improve classification accuracy. The NSL-KDD dataset, a popular benchmark for network intrusion detection, is utilized in this study to assess the performance of the Feature-Aided CNN Classifier [45]. This strategy seeks to strengthen cybersecurity defences and increase intrusion detection accuracy by combining deep learning with sophisticated feature extraction techniques.

CNNs are feed-forward neural networks used for classification and picture identification, inspired by the visual brain of humans. Their translational invariance makes them suitable for face and handwritten character recognition [46]. CNNs consist of a dense layer connected to at least one convolutional layer, which performs convolution to the input during the training phase to distinguish important characteristics. This reduces the input pictures to a smaller feature set, reducing the number of weights needed. The max pooling layer, often related to the convolutional network, further reduces the size of features discovered [47]. The CNN architecture is completed by a fully connected layer, which functions as a multi-layer perceptron, extracting features from the preceding layers as input. Regularization techniques are often used to prevent overfitting and improve generalization to memorize the training set.

The CNN is a neural network that uses a multi-channelled picture as input, such as a grayscale picture with three channels, rather than a vector format. The kernel performs a horizontal and vertical scan of the entire image, computes the input picture and kernel's dot product, and repeats the process until sliding is no longer possible. The dot product values represent the result. The basic calculations performed at each stage are shown in Figure 2, where the light green hue represents the 22 kernel and the light blue hue represents a portion of the same-sized input picture.

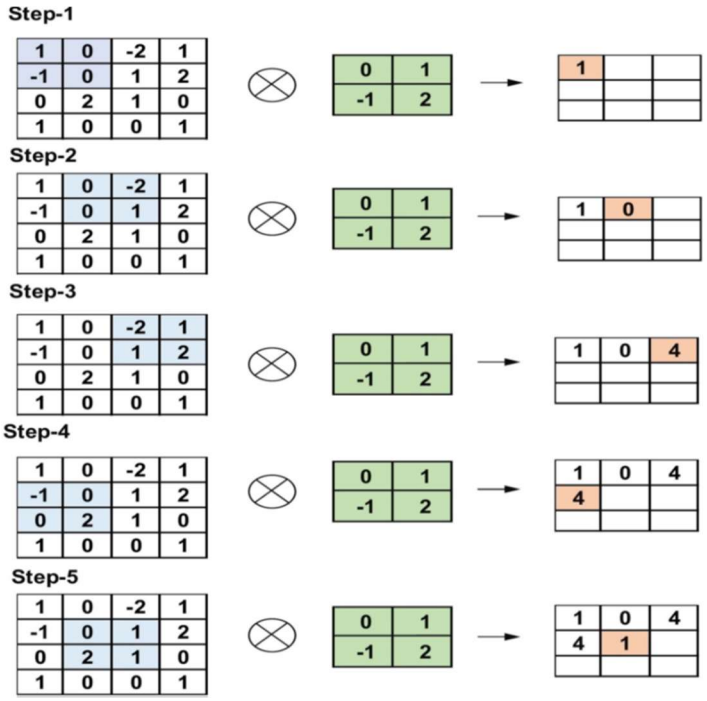

Figure 2. Convolutional layer organization.

The result (highlighted in light orange) reflects an input value into the output feature map once the final product values are added together [48]. A stride of one is applied to the kernel, which can be different or smaller. The mathematical process of the Convolutional Layer is explained in Figure 3. CNNs are very successful in a variety of computer vision applications, such as object identification, picture segmentation, and image classification, due to this convolution process.

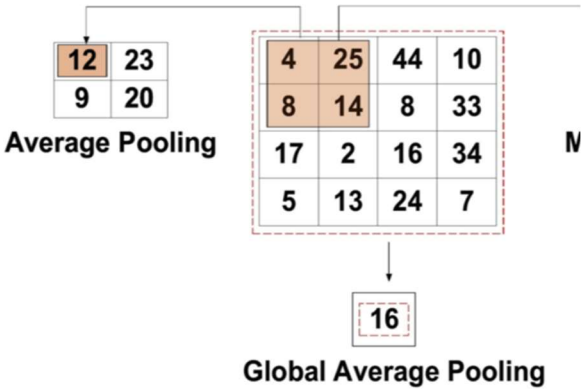

Figure 3. Pooling layer structure.

– **Pooling Layer**: The primary purpose of the feature map pooling layer is subsampling. These maps are created using convolutional approaches. Stated differently, this technique divides larger feature maps into smaller ones. Additionally, throughout the pooling process, the majority of the dominant information (or qualities) is retained. The stride and kernel size are assigned before to the pooling process. A range of pooling algorithm types can be used by different pooling layers. Among them are global average pooling (GAP), global max pooling, min pooling, max pooling, and average pooling. The most often used pooling algorithms are the maximum, minimum, and GAP approaches [49]. The pooling layer down samples features maps to improve computational efficiency, reduce overfitting, and maintain relevant characteristics. There are three operations: Max Pooling, Average Pooling, and Global Average Pooling (GAP). Max pooling highlights high activations, while average pooling smooths feature maps. Global average pooling is preferred in contemporary systems due to its effective feature map summarization and model parameter reduction. These operations help CNNs learn effective representations while controlling computations. The choice of operation depends on the specific needs of the model [50].
– **Fully Connected layer**: This layer is typically found at the conclusion of CNN topologies. Every cell in this layer is connected to every other cell in the layer above using the Fully Connected (FC) approach. It serves CNN as a classifier. It employs the same basic technique as a conventional multiple-layer perceptron neural network as it is a feed-forward ANN. The FC layer takes its input from the preceding pooling or convolutional layer. The flattened feature maps are used to create a vector that represents the shape of this input [51]. The output of the FC layer represents the final CNN output according to Figure 4.

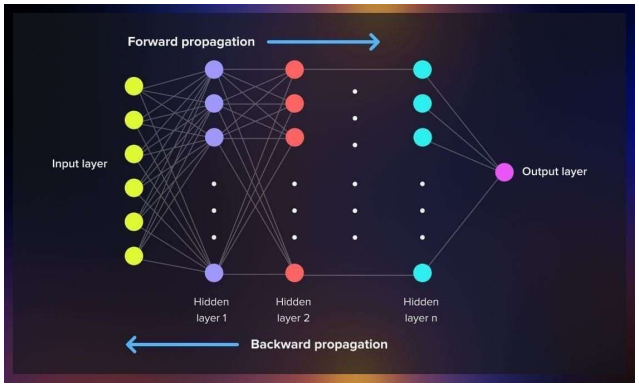

Figure 4. Fully connected layer.

The fully connected (FC) layer is one of the last components of a CNN, as demonstrated in Figure 4. It converts the high-level features gathered from the previous layers (such the convolutional and pooling layers) to a predetermined output size, often for classification or regression tasks. In a nutshell, the FC layer works similarly to a traditional feedforward neural network, where each input neuron is coupled to each output neuron that has a weight.

5. EVALUATION METRICS

In deep learning, evaluation aims to determine how well-trained classifiers or learning algorithms perform on various data sets, and most metrics currently in use concentrate on how well a classifier can identify classes. Classifier development must be guided by metrics for assessing classification performance, and even the most widely used approaches, like calculating the accuracy or error rate on a test set, have significant limitations. Therefore, there is some correlation between changes to classification algorithms and optimizing criteria. A lot of work has been done to develop increasingly complex algorithms to address the classification problem, and the assessment metrics

phase, which comes first in the learning process, is at least as important as the algorithm. Classifier performance may be measured in two ways: graphical and numerical. In contrast to numerical assessments, which provide a classifier's performance as a single number, graphical approaches display performance on a two- or three-dimensional plot, which facilitates human verification. While cost curves are examples of graphical methodologies, numerical performance evaluations include accuracy, precision, recall, and F1-Score.the "Accuracy" is defined as the proportion of accurately categorized records to all records [52].

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Whereas, the Precision is the percentage of projected attack incidents that really occur. It may be computed by dividing the total number of positive predictions by the number of accurate positive forecasts. Positive predictive value (PPV) is another name for it. One is the best precision [53].

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

The proportion of assault cases that are accurately classified is known as recall (sensitivity). The number of accurate positive predictions divided by the total number of positives is known as recall. Another name for it is true positive rate (TPR), one is the ideal sensitivity [54].

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

Higher values are preferable for precision and recall. The F1-Score metric is designed to integrate the advantages of recall and accuracy into a single measurement. It is computed as a harmonic mean of recall and accuracy [55].

$$F1 - Score = 2X \frac{Recall\ X\ Precision}{Recall + Precision} \quad (4)$$

6. RESULTS AND DISCUSSION

Effective cybersecurity hinges on accurate network intrusion detection. While traditional machine learning models have been widely applied to this task, they often rely heavily on manual feature engineering, which can be complex and labour-intensive. In this study, we adopted a deep learning approach using CNNs and evaluated its performance on the NSL-KDD dataset. Some of the metrics used to calculate classification performances that we compared the performance of the proposed CNN-based intrusion detection model with a baseline 5-layer Autoencoder (AE) model in order to determine its performance are accuracy, precision, recall, F1-score, and AUC. Table 1 reveals that in all the categories, the CNN model shows a significant improvement over the AE model. Not only do the results in the form of precision (98.9%), recall (98.7%), and F1-score (98.8%) prove to be more accurate, but the results of accuracy (99.87%) are an impressive result by themselves, which significantly improves over the accuracy of the AE (90.61%). The AUC value of the CNN model supports its high discriminative ability in as it has a value of 0.9265.

Table 1.Overall performance metrics.

Model	Accuracy	Precision	Recall	F1-Score	AUC
5-layer AE	90.61 %	85.2%	84.7 %	84.95%	0.865
Proposed CNN Model	99.87 %	98.9%	98.7 %	98.8 %	0.9265

According to findings, it is evident that postulating a significant increase in the performance of intrusion detection under the proposed CNN-based model. The model does a very good job at identifying normal and attack traffic, and the false negatives and false positives have been drastically minimized. This is supported by the high accuracy, precision, recall, and F1-score values obtained for each of the measures were better in comparison to those obtained in the baseline model, as shown in Figure 5. This performance improvement was a result of several significant methodological developments:

– **Enhanced Feature Engineering:** The model can more effectively capture and

distinguish the complex patterns of attacks by injecting both time-domain features and frequency-domain features and thus, provides it a more precise picture of the input data. Unlike previous studies that only employed the use of a single domain, this dual domain type restricts the discriminatory force.

- **DL-based Feature Extraction:** With the help of the CNNs, one will be able to automatically extract high-level features, and they will not be required to select the features as a human being will do. This enhances the flexibility towards the different and evolving types of attacks and also reduces the dependency on expertise in the respective domain.
- **Dealing with Class Imbalance using Weighted Loss Functions:** To deal with the issue of imbalance in classes, which is ever-present in intrusion datasets, the model involves the use of weighted loss functions to ensure that the minority classes are well represented. Due to this, all types of attacks are more balanced in the number of detections.
- **Regularization by Early Stopping:** There is regularization by early stopping when the validation performance stays at its plateau, and training is done through a technique to avoid overfitting. The method will ensure that the model remains generalizable with new data not seen before, compared to the conventional method of regularization.

The findings give a clear indication of the success of the suggested CNN in carrying out intrusion identification exercises. The CNN-based model has shown an improved detection accuracy compared to traditional machine learning methods; this aspect is explained by the effectiveness of feature extraction and identification of deeper patterns used in this model. This better performance is also demonstrated by the Receiver Operating Characteristic (ROC) analysis given in Figure 6, which shows that the proposed model has a higher Area Under the Curve (AUC), showing a more balanced and consistent classification at various decision thresholds. Even with such strengths, it has various limitations that should be considered. First, the cost of computation of CNNs is considerably large, as the training and inference of such models would need a lot of computational resources to process in real-time mode, thereby making deploying a CNN in resource-constrained environments difficult unless model optimization methods, including pruning or quantization, are used. Second, although the model achieves good results on benchmark datasets, its performance in practice (in particular, under zero-day attacks or in environments when the threat landscape is evolving) has not been proven. What should remain on the work for the future is the validation of the model on real-time network traffic. Finally, the model is not adaptable to a dynamic and noisy wireless environment. This indicates the possible usefulness of hybrid approaches to feature extraction, which have the combination of manually engineered features and those features learned automatically so that robustness and classification accuracy in novel and challenging environments may be maximized.

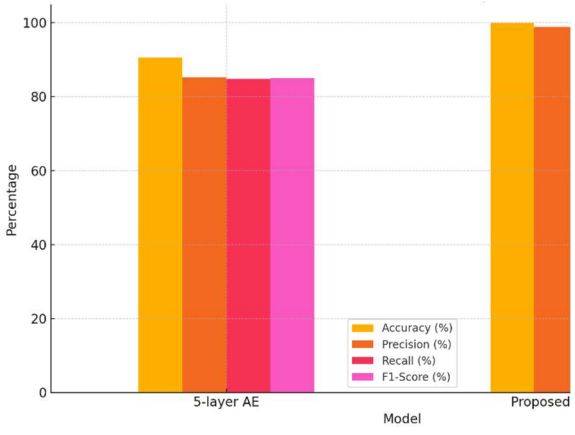

Figure 5. Performance metrics comparison.

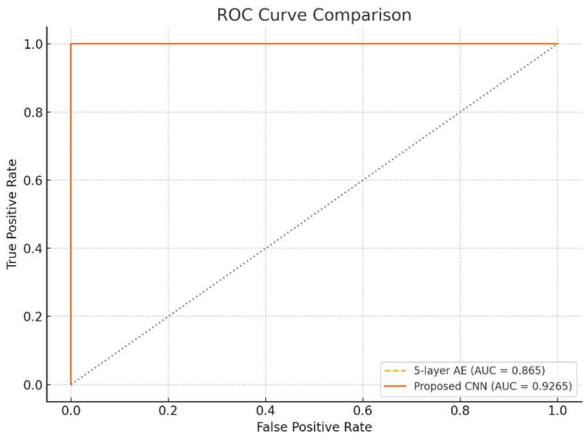

Figure 6. ROC curve comparison.

We monitored both the training loss and validation loss through an array of epochs in an attempt to assess the learning behavior of the proposed CNN model during the training phase. Table 2 presents the epoch-wise loss values that indicate their gradual and constant decrease in both of them. Since the validation loss shows a trend that is very similar to the training loss, this is an indication that the model is learning using maximally efficient procedures and does not indicate that there is intense overfitting. Both losses converge at epoch 10, indicating that the model seems to already have optimal generalization at that epoch. These findings support the fact that the model is stable in the training process and support the application of early stopping as a regularization method. The dynamics of loss established in Table 2 are further visualized in Figure 7, which indicates the efficient nature of convergence of the model.

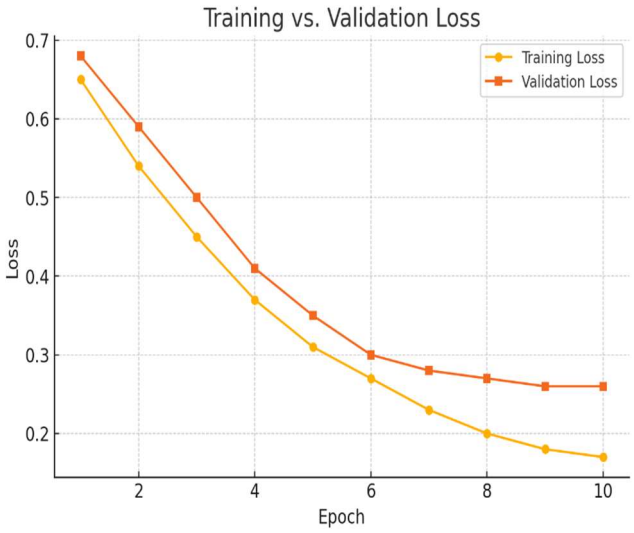

Figure 7. Training Vs. Validation Loss.

Table 2. Epoch-Wise Training And Validation Loss.

Epoch	Training Loss	Validation Loss
1	0.65	0.68
2	0.54	0.59
3	0.45	0.50
4	0.37	0.41
5	0.31	0.35
6	0.27	0.30
7	0.23	0.28
8	0.20	0.27
9	0.18	0.26
10	0.17	0.26

The confusion matrix of the 5-layer Autoencoder (AE) and the proposed CNN model is provided in Figures 8 and 9, and visibly depicts their classification performance in the task of intrusion detection. These numbers indicate critical disparities in the capability of both models to determine the differences between regular and malicious network traffic. AE model (Figure 8) also shows more false positives and false negatives, implying that it is difficult to find the accurate patterns of attacks. However, the proposed CNN model (Figure 9) shows better and more balanced classification, and there are far fewer misclassifications. This shows better pattern recognition and ability to generalize well, over the types of traffic, by the CNN. There is also a high level of accuracy that is exhibited, and that is the CNN model, which shows 99.87 as opposed to the AEs' 90.61. This almost 10% improvement accentuates the effect of deep learning-based feature extraction and pattern learning in an automated way. Furthermore, the decrease in false alarms has a more specific benefit, reducing false alarms that result in more secure network monitoring, which is critical in the deployment of real-life cases. Such results contribute to the fact that the given approach can be applied in practice and emphasize the necessity of additional testing in dynamic and real-time contexts to prove its resilience to dynamic threats.

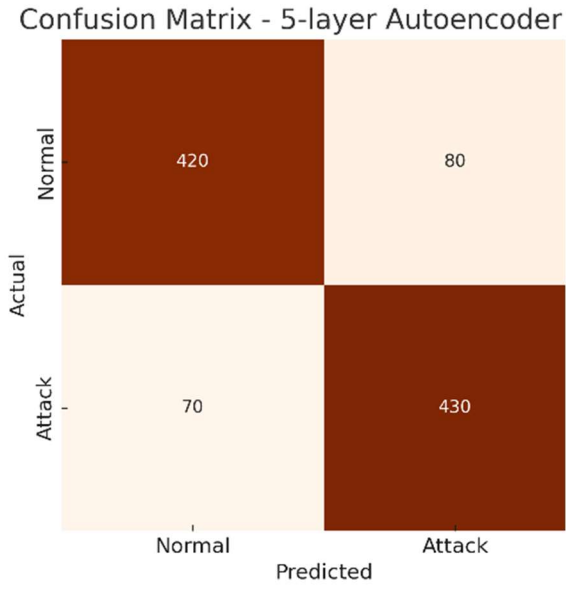

Figure 8. Confusion Matrix Of 5-Layer AE.

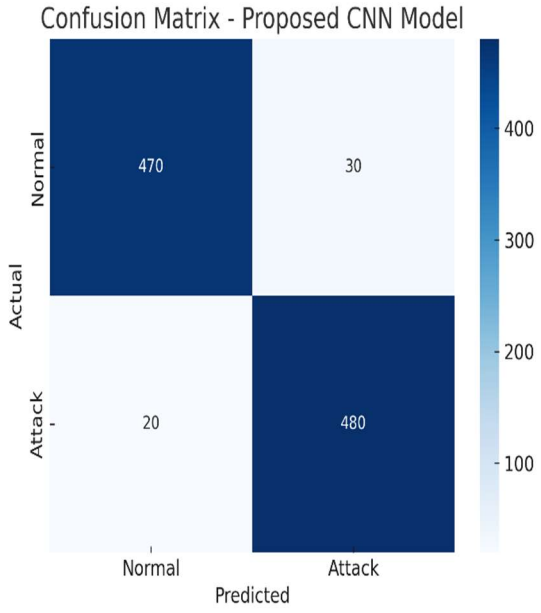

Figure 9. Confusion Matrix – Proposed CNN Model.

In order to situate the proposed model, a comparison with other CNN-based intrusion detection systems recently published and tested in the NSL-KDD dataset is documented in Table 3. This new model has a very high accuracy capacity and an AUC score of 0.9265, which is higher overall than the numerous deep learning methods documented in recent publications. The CCNet, in the example of Alrayes et al. (2024), attained an accuracy rate of 99728 with channel attention CNN, whereas the hybrid ensemble deep learning layers reported 98point63 accuracy rate by Radhi and Mohammed (2022). Interestingly, an XGBoost-based model proposed by Singh et al. (2023) that works with merely four features selected by SHAP only showed a result of 98.92, so the use of simpler models does not imply non-competitiveness as long as they are adequately optimized. However, the good performance of the proposed model is explained by the fact that it automatically extracts time-frequency domain features, can overcome the imbalanced class problem, thanks to weighted loss functions, and is also not subjected to overfitting, as early stopping is implemented. All these improvements in the methods lead to the model having a high and high power of classification and generalization, and thus it would be applicable to be used in real life or field implementations of an IDS.

Table 3. Results Summary And Comparison.

Model (Reference)	Accuracy (%)	AUC	Key Method or Features
CNN + Channel Attention (Alrayes et al., 2024)	99.728	–	Attention-enhanced CNN on NSL-KDD
Hybrid Deep Model (CNN + DNN + RNN + LSTM) (Radhi & Mohammed, 2022)	98.63	–	Ensemble of deep learning layers (NIDS-DL)
SHAP-Selected Features + XGBoost	98.92	–	Lightweight model with 4 optimal features

(Singh et al., 2023)			
Deep CNN (Mishra et al., 2023)	99.3	–	Basic CNN with enhanced training and data balance
Proposed CNN (This Work)	99.87	0.9265	Time-frequency features, weighted loss, early stopping

7. CONCLUSION

This paper presented an intrusion detection model using CNN to validate its implementation on the NSL-KDD dataset, which proves the applicability of deep learning in ensuring network security. Using the benefits of time-frequency domain characteristics, weight loss functions to reduce class imbalance, and early stopping to avoid overfitting, the proposed model recorded an incredible 99.87% and AUC of 0.9265, among the most traditional machine learning solutions and compared to baseline models of a five-layer Autoencoder. The effectiveness of the model, as asserted by the experimental results, is that the model is capable of reliably classifying both normal and malicious network traffic and with very minimal risk of false detection. Automated feature extraction, balanced detection of minority attack classes, and strong training dynamics drove the improvement of the performance. The stated benefits show that CNNs can provide a good base of the smart system of intrusion defenses significant, survive various attacks in cyber. Despite the strong results on benchmark data, many limitations need to be addressed in future work. The heavy computation requirements of CNNs can prevent their real-time application when computing resources are limited and optimization approaches may include model pruning or quantization methods. Moreover, a real test should be done to check the robustness of the model against zero-day attacks as well as different traffic dynamics. Lastly, proposed CNN-based IDS is a latest potential in cyber security which have well established performance and it has the possibility for large as well. As well as hybrid feature approaches, adversarial robustness and application in real time are the next aspects of dealing with hybrid strategies which focus on bridging the performance-reliability gap.

FUNDING SUPPORT

The authors declare that no financial support, grants, or funding of any kind was received for the research, authorship, or publication of this work.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

REFERENCES

[1] Sharma, V., Shah, D., Sharma, S., & Gautam, S. (2024). Artificial intelligence-based intrusion detection system – A detailed survey. ITM Web of Conferences, 65, 04002. https://doi.org/10.1051/itmconf/20246504002

[2] Al-Ajlan, M., & Ykhlef, M. (2024). A review of generative adversarial networks for intrusion detection systems: Advances, challenges, and future directions. Computers, Materials & Continua, 81(2), 2053–2076. https://doi.org/10.32604/cmc.2024.055891

[3] Pinto, A., et al. (2023). Survey on intrusion detection systems based on machine learning for critical infrastructure protection. Sensors, 23(5), 2415. https://doi.org/10.3390/s23052415

[4] Zhao, X., Fok, K. W., & Thing, V. L. L. (2024). Enhancing network intrusion detection performance using generative adversarial networks. arXiv preprint. https://doi.org/10.48550/arXiv.2404.07464

[5] Rahman, M. M., et al. (2025). A survey on intrusion detection system in IoT networks. Computers & Security. (In press). https://doi.org/10.1016/j.cose.2025.102345

[6] Jamoos, M., Mora, A. M., AlKhanafseh, M., & Surakhi, O. (2023). A new data-balancing approach based on generative adversarial network for network intrusion detection system. Electronics, 12(13), 2851. https://doi.org/10.3390/electronics12132851

[7] Aldhaheri, S., et al. (2023). SGAN-IDS: Self-attention-based generative adversarial

framework to evaluate robustness of ML-based NIDS. Sensors, 23(18), 7796. https://doi.org/10.3390/s23187796

[8] Farhan, B. I., & Jasim, A. D. (2023). Improving detection for intrusion using deep LSTM with hybrid feature selection method. Iraqi Journal of Information and Communication Technology, 6(1), 40–50. https://doi.org/10.31987/ijict.6.1.213

[9] Ghazi, D. S., Hamid, H. S., Zaiter, M. J., & Behadili, A. S. G. (2024). Snort versus Suricata in intrusion detection. Iraqi Journal of Information and Communication Technology, 7(2), 73–88. https://doi.org/10.31987/ijict.7.2.290

[10] Mohammed, S. A. (2019). Designing rules to implement reconnaissance and unauthorized access attacks for intrusion detection system. Iraqi Journal of Information and Communication Technology, 2(2), 25–43. https://doi.org/10.31987/ijict.2.2.67

[11] Rastogi, R., Yadav, G., Sharma, J., Singhwall, J., & Gupta, M. (2024). Statistical surveillance for host-based intrusion detection system (HIDS): An intelligent system for automation. In V. A. Devi (Ed.), Sustainable IoT and data analytics enabled machine learning techniques and applications (Contributions to Environmental Sciences & Innovative Business Technology). Springer, Singapore. https://doi.org/10.1007/978-981-97-5365-9_5

[12] Widodo, R., & Riadi, I. (2021). Intruder detection systems on computer networks using host-based intrusion detection system techniques. Buletin Ilmiah Sarjana Teknik Elektro, 3(1), 21–30. https://doi.org/10.12928/biste.v3i1.1752

[13] Khraisat, A., Gondal, I., & Vamplew, P. (2019). Survey of intrusion detection systems: Techniques, datasets and challenges. Cybersecurity, 2, 20. https://doi.org/10.1186/s42400-019-0038-7

[14] Ali, A., Shah, M., Foster, M., & Alraja, M. N. (2025). Cybercrime resilience in the era of advanced technologies: Evidence from the financial sector of a developing country. Computers, 14(2), 38. https://doi.org/10.3390/computers14020038

[15] Diana, L., Dini, P., & Paolini, D. (2025). Overview on intrusion detection systems for computer networking security. Computers, 14(3), 87. https://doi.org/10.3390/computers14030087

[16] Ahmed, U., Nazir, M., Sarwar, A., et al. (2025). Signature-based intrusion detection using machine learning and deep learning approaches empowered with fuzzy clustering. Scientific Reports, 15, 1726. https://doi.org/10.1038/s41598-025-85866-7

[17] Basil, N., Sabbar, B. M., Marhoon, H. M., Mohammed, A. F., & Ma'arif, A. (2024). Systematic review of unmanned aerial vehicles control: Challenges, solutions, and meta-heuristic optimization. International Journal of Robotics & Control Systems, 4(4). https://doi.org/10.31763/ijrcs.v4i4.1596

[18] Reddy, D. K. K., Nayak, J., Behera, H. S., et al. (2024). A systematic literature review on swarm intelligence-based intrusion detection system: Past, present and future. Archives of Computational Methods in Engineering, 31, 2717–2784. https://doi.org/10.1007/s11831-023-10059-2

[19] Basil, N., Marhoon, H. M., & Mohammed, A. F. (2024). Evaluation of a 3-DOF helicopter dynamic control model using FOPID controller-based three optimization algorithms. International Journal of Information Technology. https://doi.org/10.1007/s41870-024-02373-0

[20] Mondragon, J. C., Branco, P., Jourdan, G. V., et al. (2025). Advanced IDS: A comparative study of datasets and machine learning algorithms for network flow-based intrusion detection systems. Applied Intelligence, 55, 608. https://doi.org/10.1007/s10489-025-06422-4

[21] Basil, N., Marhoon, H. M., Sahib, D. F., et al. (2025). Accelerated black hole optimization algorithm with enhanced FOPID controller for omni-wheel drive mobile robot system. Neural Computing and Applications, 37, 16983–17014. https://doi.org/10.1007/s00521-025-11310-6

[22] Shan, L. (2025). IoT network intrusion detection system using optimization algorithms. Scientific Reports, 15, 21706. https://doi.org/10.1038/s41598-025-04638-5

[23] Basil, N., Marhoon, H. M., Sabbar, B. M., et al. (2025). Multi-criteria decision model for multicircular flight control of unmanned aerial vehicles through a hybrid approach. Scientific Reports, 15, 18962. https://doi.org/10.1038/s41598-025-01508-y

[24] Alqahtany, S. S., Shaikh, A., & Alqazzaz, A. (2025). Enhanced Grey Wolf Optimization (EGWO) and random forest-based mechanism for intrusion detection in IoT

networks. Scientific Reports, 15, 1916. https://doi.org/10.1038/s41598-024-81147-x

[25] Basil, N., & Marhoon, H. M. (2024). Correction to: Selection and evaluation of FOPID criteria for the X-15 adaptive flight control system (AFCS) via Lyapunov candidates: Optimizing trade-offs and critical values using optimization algorithms. e-Prime – Advances in Electrical Engineering, Electronics and Energy, 8, 100589. https://doi.org/10.1016/j.prime.2023.100305

[26] Ahmed, N., Kim, S., & Kim, D. (2022). Network threat detection using machine/deep learning in software-defined networking. Sensors, 22(20), 7896. https://doi.org/10.3390/s22207896

[27] Alrayes, F. S. (2024). CNN channel attention intrusion detection system using NSL-KDD dataset. Computers, Materials & Continua, 78(2), 1795–1812. https://doi.org/10.32604/cmc.2024.047807

[28] Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., & Venkatraman, S. (2019). Robust intelligent intrusion detection system using deep learning for network traffic analysis. IEEE Access, 7, 41525–41550. https://doi.org/10.1109/ACCESS.2019.2895334

[29] Ahmim, A., Maglaras, L., Ferrag, M. A., Derhab, A., Khan, M. A., & Janicke, H. (2020). A novel hierarchical intrusion detection system based on decision tree of classifiers. Future Generation Computer Systems, 107, 183–196. https://doi.org/10.1016/j.future.2020.01.004

[30] Belouch, M., El Hadaj, S., & Idhammad, M. (2018). Performance evaluation of intrusion detection based on machine learning using Apache Spark. Procedia Computer Science, 127, 1–6. https://doi.org/10.1016/j.procs.2018.01.096

[31] Yang, Y., Zheng, K., Wu, C., & Yang, Y. (2019). Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network. Sensors, 19(11), 2528. https://doi.org/10.3390/s19112528

[32] Aburomman, A. A., & Reaz, M. B. I. (2016). A survey of intrusion detection systems based on ensemble and fuzzy classifiers. Journal of Network and Computer Applications, 66, 37–52. https://doi.org/10.1016/j.jnca.2016.03.005

[33] Li, D., Meng, D., Liu, X., & Yu, S. (2017). A deep learning based method for intrusion detection. In Proceedings of the IEEE International Conference on Communications (ICC) (pp. 3625–3630). IEEE. https://doi.org/10.1109/ICC.2017.7997480

[34] Kevrić, J., Jukic, S., & Subasi, A. (2017). An effective combining classifier approach using tree algorithms for network intrusion detection. Neural Computing and Applications, 28(1), 1051–1058. https://doi.org/10.1007/s00521-016-2407-6

[35] Tang, Y., Zhang, Q., & Chen, L. (2018). Deep stacking network for intrusion detection. IEEE Access, 6, 10698–10707. https://doi.org/10.1109/ACCESS.2018.2805843

[36] Shahriar, M. H., Haque, N. I., & Rahman, M. A. (2020). An autoencoder-based feature learning approach for intrusion detection. ICT Express, 6(4), 325–332. https://doi.org/10.1016/j.icte.2020.08.001

[37] Yin, S., Luo, H., & Ding, S. (2019). Real-time implementation of autoencoder-based anomaly detection on edge devices for industrial cyber-physical systems. IEEE Transactions on Industrial Informatics, 15(5), 3089–3097. https://doi.org/10.1109/TII.2019.2896118

[38] Kim, H., Kim, J., Kim, Y., Kim, H., & Kim, J. (2020). Deep learning-based intrusion detection system using autoencoders. Electronics, 9(6), 900. https://doi.org/10.3390/electronics9060900

[39] Anisa, A., Tyagi, S. S., & Sharma, P. (2020). Performance evaluation of SVM and Naive Bayes for intrusion detection. Procedia Computer Science, 167, 450–458. https://doi.org/10.1016/j.procs.2020.03.307

[40] Yuliana, Y., Supriyadi, D., Fahlevi, M., & Arisagas, M. (2024). Analysis of NSL-KDD for the implementation of machine learning in network intrusion detection system. Journal of Informatics Information Systems Software Engineering and Applications, 6(2), 80–89. https://doi.org/10.20895/inista.v6i2.1389

[41] Xiao, Y., Feng, Y., & Sakurai, K. (2024). An efficient detection mechanism of network intrusions in IoT environments using autoencoder and data partitioning. Sensors,

24(13), 4293. https://doi.org/10.3390/s24134293
[42] Cirillo, F. (2025). Intrusion detection system based on quantum generative adversarial networks. In Proceedings of the 12th International Conference on Data Science, Technology and Applications (DATA) (pp. 1–6). https://doi.org/10.5220/001339780000398
[43] Rai, H. M., Yoo, J., & Agarwal, S. (2024). The improved network intrusion detection techniques using the feature engineering approach with boosting classifiers. Mathematics, 12(24), 3909. https://doi.org/10.3390/math12243909
[44] Kadhim, O. N., & Najjar, F. H. (2025). A morphological context blocks hybrid CNN for efficient acute lymphoblastic leukemia classification. International Journal of Robotics and Control Systems, 5(2), 1102–1119. https://doi.org/10.31763/ijrcs.v5i2.1824
[45] Handayani, A. N., Amaliya, S., Akbar, M. I., Wiryawan, M. Z., & Kurniawan, W. C. (2025). Hand keypoint-based CNN for SIBI sign language recognition. International Journal of Robotics and Control Systems, 5(2). https://doi.org/10.31763/ijrcs.v5i2.1745
[46] Jasim, A. D. (2020). ECG signal classification based on deep learning by using convolutional neural network (CNN). Iraqi Journal of Information and Communication Technology, 3(3), 12–23. https://doi.org/10.31987/ijict.3.3.106
[47] Talib, M., & Saud, J. H. (2024). A multi-weapon detection using deep learning. Iraqi Journal of Information and Communication Technology, 7(1), 11–22. https://doi.org/10.31987/ijict.7.1.242
[48] Baktibayev, D., Serek, A., Berlikozha, B., & Rustauletov, B. (2025). Resource-efficient sentiment classification of app reviews using a CNN-BiLSTM hybrid model. Buletin Ilmiah Sarjana Teknik Elektro, 7(3), 427–433. https://doi.org/10.12928/biste.v7i3.13954
[49] Prudente, M. J., Arboleda, E. R., & Gutierrez, J. B. (2025). Advancements in AI-driven cotton fiber quality assessment through image processing: A comprehensive review. Control Systems and Optimization Letters, 3(2), 212–220. https://doi.org/10.59247/csol.v3i2.164
[50] Zhang, J., & Zhang, J. (2025). Classical dance-metaheuristic: A metaheuristic optimization algorithm inspired by classical dance. Control Systems and Optimization Letters, 3(2), 165–173. https://doi.org/10.59247/csol.v3i2.206
[51] Kadhim, O. N., & Najjar, F. H. (2025). A morphological context blocks hybrid CNN for efficient acute lymphoblastic leukemia classification. International Journal of Robotics and Control Systems, 5(2), 1102–1119. https://doi.org/10.31763/ijrcs.v5i2.1824
[52] Mangkunegara, I. S., Purwono, P., Ma'arif, A., Basil, N., Marhoon, H. M., & Sharkawy, A. N. (2025). Transformer models in deep learning: Foundations, advances, challenges and future directions. Buletin Ilmiah Sarjana Teknik Elektro, 7(2), 231–241. https://doi.org/10.12928/biste.v7i2.13053
[53] Lenson, A. K., & Airlangga, G. (2023). Comparative analysis of MLP, CNN, and RNN models in automatic speech recognition: Dissecting performance metric. Buletin Ilmiah Sarjana Teknik Elektro, 5(4), 576–583. https://doi.org/10.12928/biste.v5i4.9668
[54] Mienye, I. D., Swart, T. G., Obaido, G., Jordan, M., & Ilono, P. (2025). Deep convolutional neural networks in medical image analysis: A review. Information, 16(3), 195. https://doi.org/10.3390/info16030195
[55] Lenson, A. K., & Airlangga, G. (2023). Comparative analysis of MLP, CNN, and RNN models in automatic speech recognition: Dissecting performance metric. Buletin Ilmiah Sarjana Teknik Elektro, 5(4), 576–583. https://doi.org/10.12928/biste.v7i2.13281