**Turkish Journal of Engineering**

https://dergipark.org.tr/en/pub/tuje

**e-ISSN 2587-1366**

# Comparative Study of BiGRU with Multi-Head Attention and CNN for Network Intrusion Detection Using a Cleaned and Balanced CSE-CIC-IDS 2018 Dataset

**Suresh Kumar Balasubramanian** [1], **Senthilkumar Perumal** [*2]

[1]*Assistant Professor, Department of Computer Science, Government Arts and Science College, Sriperumbudur, Chennai. Tamil Nadu, India, sureshaucis@gmail.com*

[2]*Assistant Professor, Department of Computer & Information Science, Annamalai University, Tamil Nadu, India, senthil.sp74@gmail.com*

Cite this study: Suresh Kumar, B., & Senthilkumar, P. (2025). Comparative study of BiGRU with multi-head attention and CNN for network intrusion detection on a cleaned and balanced CSE-CIC-IDS 2018 dataset. *Turkish Journal of Engineering, 9(4), 725-737.*

https://doi.org/10.31127/tuje.1695208

**Keywords**
Network intrusion detection, BiGRU with Attention, Convolutional neural networks (CNN), CSE-CIC-IDS2018 Dataset, Cybersecurity

**Research Article**

Received:08.05.2025
Revised:08.09.2025
Accepted:08.09.2025
Published:30.10.2025

**Abstract**

With the age of advanced cyber attacks, robust intrusion detection systems are inevitable in order to protect the network from insecurity. This work presents a new comparative performance evaluation of two deep learning models, namely, Bidirectional Gated Recurrent Unit with Multi Head Attention (BiGRU + MHA) and Convolutional Neural Network (CNN), on the updated CSE-CIC-IDS 2018 dataset (Version 1, 2024). The data set was cleaned and balanced meticulously by eliminating duplicate entries and a two-stage resampling method with random undersampling accompanied with synthetic minority oversampling for accurate representation of both frequent as well as infrequent types of attacks. The experimental results confirm that both models provided superior detection performance, with BiGRU + MHA consistently outperforming CNN. Specifically, BiGRU + MHA provided 99.65 percent accuracy as well as ROC AUC of 99.71 percent, whereas CNN provided 98.85 percent accuracy as well as ROC AUC of 98.92 percent. The observations identify the advantage of using the combination of temporal sequence modeling as well as attention for identifying advanced intrusion patterns in network traffic. Generally, the results confirm that the use of deep temporal learning in combination with structured preparation of the data holds the capability for leading to highly effective intrusion detection, with great potential for strengthening cybersecurity solutions.

## Introduction

Network intrusion detection systems play an important role in securing computer networks against a wide range of malicious activities. Traditional signature-based systems fail in detection of novel or emerging attacks. Overcoming this weakness, machine learning as well as deep learning techniques have proven efficient alternatives in learning dynamically how to distinguish between malicious and normal traffic patterns. Recent advances in artificial intelligence and machine learning have demonstrated remarkable success across various engineering

domains, establishing neural networks as robust architectures for pattern recognition and classification tasks [44]. Yang et al. presented a BiGRU-Inception-CNN model with attention, hybrid sampling, and feature selection for enhancing IIoT intrusion detection over complex as well as imbalanced datasets [1]. Hu et al. presented a SAG-BiGRU model employing self-attention as well as resampling for enhancing intrusion detection accuracy, especially over imbalanced datasets such as CICIDS2017 as well as NSL-KDD [2]. Song et al. presented TGA, an intrusion detection hybrid model that combines TCN, BiGRU, as well as self-attention for both local as well as global temporal features, with 97.83 accuracy for CSE-CIC-IDS2018 [3]. Wang et al. tested six deep learning models over CSE-CIC-IDS2018, finding that individual DNN, RNN, as well as CNN models provided high accuracy with increased efficiency in inference over combined models [4]. Alzughaibi as well as El Khediri implemented DNN-based IDS models with MLP as well as backpropagation as well as with PSO, achieving over 98% for binary as well as multi-class intrusion detection in environments of clouds [5]. Cao et al. presented intrusion detection-based CNN-GRU with hybrid sampling as well as attention mechanisms, achieving high accuracy in multiple datasets with efficient handling of class imbalance [6]. Kanimozhi as well as Jacob performed classifier-based comparison for botnet detection using the CSE-CIC-IDS2018 dataset, which showed that AI-based methods performed better in accuracy as well as calibration when compared with traditional models [7]. Cao et al. presented intrusion detection-based CNN-BiGRU with hybrid sampling as well as feature selection, achieving enhanced accuracy in multiple benchmark datasets [8]. Udurume et al. performed comparison of traditional ML models with the CNN-BiLSTM-based deep learning-based intrusion detection with traditional models for detection in the Internet of Things, wherein CNN-BiLSTM performed with highest accuracy over NSL-KDD as well as UNSW-NB15 datasets [9]. Zhang et al. presented an enhanced BiLSTM with multi-head attention for enhancing intrusion detection accuracy over high-dimensional as well as imbalanced datasets, with over 95 accuracy for three benchmark datasets [10]. Guo and Xie developed the TRBMA model that combines 1D-ResNet, TCN, BiGRU, and Multi-Head Attention in order to enhance temporal feature learning as well as improve classification accuracy. The advanced variant, namely, TRBMA (BS-OSS), adopts hybrid sampling for detection of minority types of attacks with improved accuracy up to 99.88% with the CIC-IDS-2017 dataset [11]. Susilo et al. presented an intrusion detection system in IoT settings involving autoencoders, LSTM network, as well as multistage feature extraction using CNN for intrusion detection. [12]. Aljabri presented an effective intrusion detection system with an optimized IWSO for IoT settings involving integration of the Bidirectional GRU with Multi-Head Attention (BiGRU-MHA). The system evaluated with Edge-IIoT dataset produced 98.28% classification accuracy [13]. Wang et al. presented an intrusion detection system involving the integration of CNN-BiGRU capable of extracting both spatial as well as temporal patterns for enhancing intrusion detection accuracy as well as suppressing false alarms [14]. Hu et al. developed a CNN-KOA-BiGRU model that accurately detects APT attacks by combining deep learning with an optimization algorithm to enhance feature extraction and classification [15]. Hewapathirana introduced a two-stage intrusion detection framework using SAE and Spark-based approaches, showing SAE's superior accuracy and Spark's strength in real-time efficiency [16]. Li et al. proposed ADFCNN-BiLSTM, combining deformable convolution, BiLSTM, and attention mechanisms to improve intrusion detection across spatial and temporal features [17]. Zhang et al. reviewed deep learning applications in IDS, highlighting key challenges in spatiotemporal feature extraction and data imbalance, and suggested future research directions [18]. Deshmukh and Ravulakollu introduced IIDNet, optimized CNN-based IDS for IoT, achieving high accuracy and reduced training time on the UNSW-NB15 dataset [19]. El-Shafeiy et al. proposed DCGR_IoT, a deep learning-based IDS combining CNN and CGRN to achieve 99.2% accuracy in detecting IoT network intrusions [20]. Attack et al. (2025) developed an ensemble model using FA-CNN and autoencoders, achieving strong detection rates on NSL-KDD and CICIDS2017, especially for rare attacks like U2R and Heartbleed [21]. Han and Pak (2023) demonstrated that using entire session packet data with a hierarchical LSTM significantly boosts intrusion detection accuracy [22]. Imrana et al. (2024) introduced CNN-GRU-FF, a fusion-based intrusion detection model that effectively handles class imbalance and achieves high detection rates on benchmark datasets [23]. Xin et al. (2018) emphasized that RNNs are well-suited for sequential data, while CNNs efficiently reduce model complexity using weight sharing, making them ideal for tasks like image and speech recognition [24]. The CSE-CIC-IDS 2018 dataset is a

contemporary benchmark for intrusion detection, replicating varied attack situations against realistic network environments. Version 1 of the dataset, published in February 2024, is an improvement over previous versions by providing recent, cleaned traffic records. While like most raw intrusion detection data it is still plagued with problems such as duplicate records, missing and malformed fields, and a heavy imbalance between normal and attack classes. To overcome these shortcomings, we used a systematic preprocessing pipeline. Duplicates and erroneous records were eliminated, and the class imbalance was resolved using a two-stage resampling technique. It entailed undersampling the majority classes using random undersampling and oversampling minority classes using synthetic oversampling with SMOTE. The preprocessed data offers a more balanced training set, which is necessary for creating unbiased and efficient models. Deep architectures have been observed to exhibit robust performance in intrusion detection. Convolutional neural networks are trained on hierarchical representations of raw data with little human intervention and are therefore particularly capable of detecting spatially localized patterns of trafficThe effectiveness of deep convolutional neural networks has been demonstrated across multiple domains for feature extraction and pattern recognition, making them particularly suitable for complex classification tasks [45]. Recurrent models, like the gated recurrent unit, have the capability to learn temporal relationships of time series data. Bidirectional GRUs, specifically, read sequences in both directions, drawing context from past and future packets. When paired with multi head attention mechanisms, these models acquire the capacity to pay attention to the most informative parts of a sequence, enhancing their sensitivity to subtle and long term patterns in network traffic. While both convolutional and recurrent attention based models have shown individual robust performance, head to head comparisons between them on the same well processed datasets are still few. In this research, we systematically compare a bidirectional GRU with multi head attention with a baseline CNN, on the same cleaned and resampled CSE CIC IDS2018 dataset. Both models are trained end to end to classify network traffic into several classes, including benign and several types of attacks. Our contributions are a careful comparative analysis of these two architectures and a demonstration that the BiGRU with multi head attention performs very high classification performance under controlled data conditions consistently. We also highlight the essential role of preprocessing in achieving these results, especially in dealing with noise and imbalance. The rest of this paper is structured as follows. Section 2 explains the dataset and preprocessing techniques. Section 3 defines the model architectures and training processes. Section 4 discusses the experimental results and comparative assessment. Section 5 concludes with a discussion of important findings and future research directions.

## Dataset Preparation

### Dataset consolidation

Mohamed (2024) published the preprocessed and balanced CSE-CIC-IDS2018 dataset, providing a more polished benchmark for testing intrusion detection systems. [25]. Dataset preparation began with systematically extracting and arranging the CSV files. Each file, corresponding to a specific traffic situation, was validated and then combined into a single, unified dataset. This combining reduced fragmentation, ensured consistency, and allowed for easier downstream processing for machine learning purposes.

### Preprocessing challenges

Before preprocessing, the dataset also contained several quality problems that could hinder uniform model training. They were duplicate records, missing information, inconsistent formats, unprocessed categorical variables, and extreme class imbalance. The original dataset consisted of approximately 9.6 million records, the majority of which introduced noise or unreliability. For enabling robust analysis, a strict data cleaning pipeline was required to improve overall data integrity.

### Data cleaning and deduplication

The cleaning activity was centered on improving the dataset's reliability and consistency. Duplicate records were dropped to avoid data leakage, missing or corrupted values were imputed or dropped, depending on the severity. Inconsistent records containing irrelevant content or incorrect formatting were also dropped to guarantee the end dataset included just valid and structured traffic data. Following this cleaning process, the dataset was shrunk to 5,183,021 records, dramatically enhancing its quality and readiness for model training.

**Table 1.** Data counts before and after cleaning

| Attack Type | Initial Data Distribution - Sample Count | Data Distribution after Cleaning and Deduplication |
|---|---|---|
| Benign | 6876913 | 3830384 |
| DDoS attack-HOIC | 686012 | 575364 |
| DDoS attacks-LOIC-HTTP | 576191 | 198861 |
| DoS attacks-Hulk | 461912 | 145199 |
| Bot | 286191 | 144535 |
| FTP-BruteForce | 193360 | 140610 |
| SSH-Bruteforce | 187589 | 94048 |
| Infiltration | 161934 | 41406 |
| DoS attacks-SlowHTTPTest | 139890 | 9908 |
| DoS attacks-GoldenEye | 41508 | 1730 |
| DoS attacks-Slowloris | 10990 | 555 |
| DDoS attack-LOIC-UDP | 1730 | 228 |
| Brute Force -Web | 611 | 84 |
| Brute Force -XSS | 230 | 55 |
| SQL Injection | 87 | 54 |
| Total | 9625148 | 51,83,021 |

This drop shows the dramatic improvement in data consistency and purity, providing a strong foundation for accurate and unbiased model training.

**Class imbalance handling**

Class imbalance is a recurring problem with intrusion detection datasets and usually results in models with poor performance on minority attack types. Even though the CSE–CIC-IDS2018 dataset is commonplace, there are few studies that have seriously tackled this issue [26]. Kamal and Mashaly [27] illustrated that the hybrid models like Transformer–CNN can gain greatly from resampling strategies such as SMOTE, ADASYN, and class weight. Similarly, Buda et al. [28] explored how imbalance skews deep learning models, while Abd Elrahman and Abraham [29] argued that no single resampling strategy fits all cases. To tackle this challenge, we adopted a two-phase resampling strategy. First, Random Undersampling (RUS) was used to reduce overrepresented classes to a maximum of 100,000 records each. Then, the Synthetic Minority Over-sampling Technique (SMOTE) was used to increase minority class samples without repeating current entries. The choice of 10,000 samples for minority classes was determined through empirical testing, balancing between providing sufficient representation for model learning and maintaining computational feasibility. This threshold ensures each minority class has adequate training samples (≥10,000) while preventing excessive computational overhead during training. For example, the Brute Force Attack class, which initially had only 837 instances, was increased to 10,000 varied synthetic instances. Lastly, label encoding transformed categorical class names into numerical labels to enable supervised learning, a step suggested by Fernández et al. [30] to enhance model efficiency.

**Table 2.** Class distribution after resampling

| Category | Grouped Category Counts | After RUS Count | After SMOTE Count | Numeric Label |
|---|---|---|---|---|
| NORMAL | 3830384 | 100000 | 100000 | 0 |
| DoS/DDoS Attack | 972523 | 100000 | 100000 | 1 |
| Botnet Activity | 144535 | 100000 | 100000 | 2 |
| Brute Force Attack | 837 | 837 | 10000 | 3 |
| Infiltration & Exploits | 140694 | 100000 | 100000 | 4 |
| SSH-Brute Force | 94048 | 94048 | 100000 | 5 |

The balanced distribution obtained ensures the models learn from both frequent and infrequent classes of attacks well, enhancing generalization and accuracy of detection. Furthermore, we conducted feature importance analysis with Random Forest feature importance scores to determine the most discriminative features. We selected the top 20 features that contribute to intrusion detection on the basis of importance scores, which reduced dimensionality while preserving classification performance. We also investigated Principal Component Analysis (PCA) as another dimensionality reduction approach, but feature selection using importance scores performed better in terms of interpretability and performance preservation. Korkmaz and Şahin (2024) demonstrated that proper feature selection techniques can significantly enhance intrusion detection performance while reducing computational complexity, supporting our approach to feature importance-based selection [31].

## Methodology

This part describes the architecture, training procedure, and testing of two deep learning models for multi-class network intrusion detection: a Bidirectional Gated Recurrent Unit model with Multi-Head Attention (BiGRU+MHA), and a Convolutional Neural Network (CNN). Both models strive to correctly separate benign traffic from different types of malicious attacks using spatial and temporal features of the input data.

### Overview of models

The models were chosen to demonstrate two alternative approaches to sequence modeling: one addressing temporal dynamics with recurrent units and attention mechanisms, and the other addressing spatial dependencies with convolutional operations. Both networks were adapted to accept one-dimensional time-series input following preprocessing of the network traffic data. Erdoğan et al. (2024) conducted comprehensive comparisons of various deep learning architectures for network security applications, highlighting the importance of architectural choices in achieving optimal performance [32]. Input sequence length was set to 100 time steps after careful examination of different lengths (50, 100, 150, 200). The parameter was tuned using grid search validation, where sequence length 100 provided the optimal balance between temporal dependency capture and computational expense. Reducing sequence length (≤50) could not capture long-term attack patterns, and increasing sequence length (≥150) increased training time with minimal improvement in performance. Vaswani et al. (2017) introduced the Transformer architecture, a breakthrough model based entirely on attention mechanisms, which outperformed existing models in machine translation while enabling faster training and greater parallelization [33]. Benchama et al. (2024) introduced a hybrid CNN-BiGRU model optimized with Optuna and enhanced by SMOTE to address data imbalance in NIDS, achieving 98.83% accuracy on the NSL-KDD dataset while effectively detecting minority class intrusions [34]. Yang et al. (2024) proposed an advanced intrusion detection approach for Industrial IoT by integrating attention mechanisms, BiGRU, and Inception-CNN, coupled with hybrid sampling and feature selection techniques, achieving improved detection rates on datasets like Edge-IIoTset and CIC-IDS2017 [35]. The initial model employs recurrent units and attention to model temporal relationships, whereas the second employs convolution operations to model spatial relationships. Each of them is designed to take one-dimensional time-series data. This bringing together of methods supports intrusion detection

by leveraging both time and spatial attributes for higher accuracy.

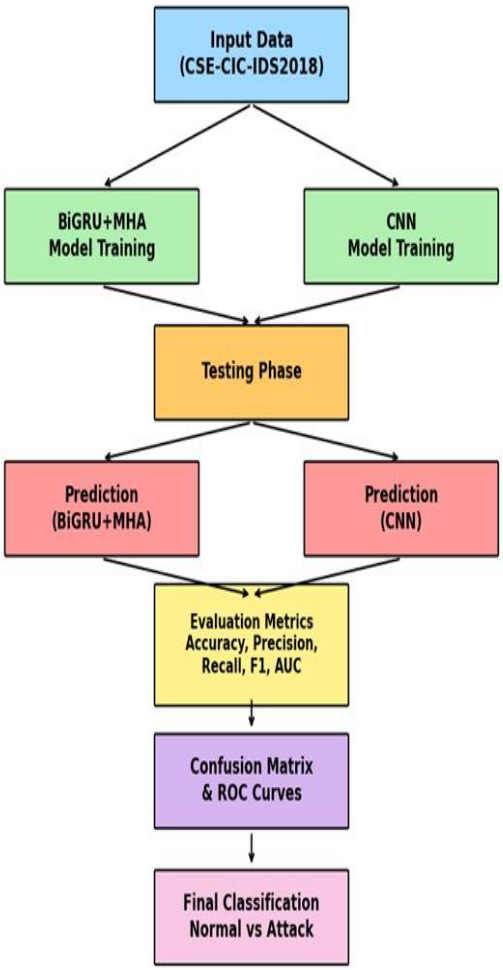


**Figure 1.** Workflow of BiGRU+MHA and CNN models for intrusion detection using the CSE-CIC-IDS2018 dataset.

### Bigru with multi-head attention

Recurrent Neural Networks (RNNs) are inherently well-formulated to address sequential dependencies but conventional variants struggle with long dependencies. Neural network architectures have proven their versatility in modeling complex relationships across diverse applications, making them particularly effective for sequential data processing tasks [46]. Gated Recurrent Units (GRUs) circumvent this problem using gating mechanisms in managing information flow and retention. In this study, a Bidirectional GRU (BiGRU) is utilized to capture context from preceding and succeeding time steps.

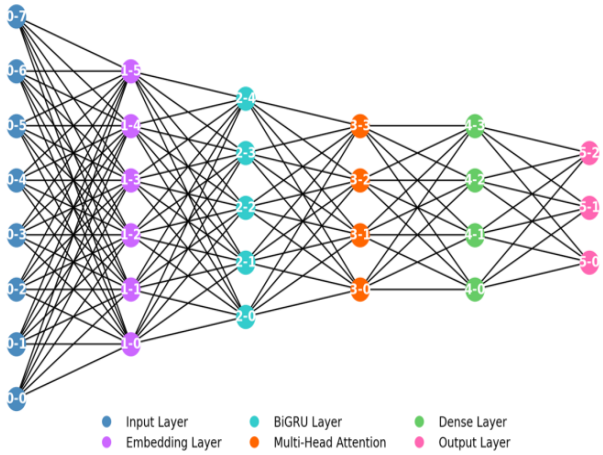


**Figure 2.** BiGRU + Multi-head attention (MHA) architecture for intrusion detection

To enhance the representations learned, an MHA layer is stacked over the BiGRU output. This allows the model to weigh different regions of the sequence together, making the model better capable of recognizing sophisticated patterns. The multi-head attention uses 8-sized 64 attention heads to allow the model to examine different patterns simultaneously. Every attention head is trained to examine different sections of the input sequence, and these sections contribute complementary information to allow a general pattern to be detected. It is then processed through a global average pooling layer, fully connected layers, and a final softmax layer for prediction.

### Mathematical notation for bigru + multi-head attention model

Let the input feature sequence be denoted by $\boldsymbol{X} \in \boldsymbol{R}^{\boldsymbol{T}\times\boldsymbol{F}}$ , where $\boldsymbol{T}$ is the number of timesteps and $\boldsymbol{F} = \boldsymbol{1}$ is the number of input features per timestep (after reshaping). The model processes the input through the following stages:

Bidirectional GRU Layer

$$\boldsymbol{H} = \boldsymbol{BiGRU}(\boldsymbol{X}) \in \boldsymbol{R}^{\boldsymbol{T}\times\boldsymbol{2d}} \quad (1)$$

Where $\boldsymbol{d}$ is the number of hidden units per direction (here $\boldsymbol{d}$ = 48)

Dropout and Layer Normalization :

$$\boldsymbol{H}' = \boldsymbol{LayerNorm}(\boldsymbol{Droput}(\boldsymbol{H})) \quad (2)$$

Multi-Head Self Attention:

$$\boldsymbol{A} = \boldsymbol{MHA}(\boldsymbol{H}', \boldsymbol{H}', \boldsymbol{H}') \in \boldsymbol{R}^{\boldsymbol{T}\times\boldsymbol{2d}} \quad (3)$$

Residual Connection with Activation:

$$\boldsymbol{R} = \boldsymbol{LayerNorm}(\boldsymbol{H}' + \boldsymbol{ReLU}(\boldsymbol{Dropout}(\boldsymbol{A}))) \quad (4)$$

Global Average Pooling :

$$\boldsymbol{v} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{R_t} \in \boldsymbol{R^{2d}} \quad (5)$$

Fully Connected Layers and Output:

$$\boldsymbol{z} = \boldsymbol{dropout}\left(\boldsymbol{ReLU}\left(\boldsymbol{W_1}\,\boldsymbol{v} + \boldsymbol{b_1}\right)\right) \in \boldsymbol{R^{2d}} \quad (6)$$

$$\hat{\boldsymbol{y}} = \boldsymbol{Softmax}(\boldsymbol{W_2}\,\boldsymbol{z} + \boldsymbol{b_2}) \in \boldsymbol{R^{C}} \quad (7)$$

Here, ŷ represents the predicted class probabilities, and C is the number of output classes.

## Convolutional neural network

While CNNs are widely used in image processing, they can actually be used on time-series data as well with local temporal learning. Machine learning classification techniques have demonstrated consistent performance across various pattern recognition applications, with convolutional architectures showing particular effectiveness in feature extraction tasks [47]. We employ a stacked CNN structure of a 1D CNN for our model, where hierarchical features in network traffic sequences are learned using multiple convolutional layers. Each convolutional block has ReLU activation, max-pooling to decrease dimension and increase robustness, and dropout layers to prevent overfitting. The last feature maps are flattened and then passed through fully connected layers before providing class probabilities through a softmax output.

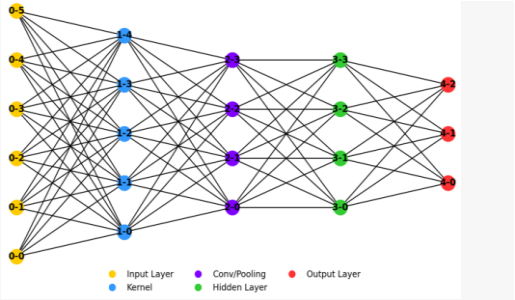


**Figure 3:** Convolutional neural network (CNN) architecture for intrusion detection

## Mathematical notation for convolutional neural network model

The same input sequence $X = \in R^{T \times F}$ is processed as follows:

First Convolution +Pooling

$$C_1 = ReLU\left(Conv1D_{k\,=\,3,c=64}(X)\right) \in R^{T \times 64} \quad (8)$$

$$P_1 = MaxPool1D\left(C_1, P = 2\right) \in R^{T/2 \times 64} \quad (9)$$

Second Convolution + Pooling:

$$\boldsymbol{C_2} = \boldsymbol{ReLU}\left(\boldsymbol{Conv1D_{k=3,c=128}}(\boldsymbol{P_1})\right) \boldsymbol{R^{\frac{T}{2} \times 128}} \quad (10)$$

$$\boldsymbol{P_2} = \boldsymbol{MaxPool1D}\left(\boldsymbol{C_2}, \boldsymbol{p} = \boldsymbol{2}\right) \in \boldsymbol{R^{T/4 \times 128}} \quad (11)$$

Flatten and Fully Connected Layers:

$$\boldsymbol{f} = \boldsymbol{Flatten}(\boldsymbol{P_2}) \in \boldsymbol{R^{128.T/4}} \quad (12)$$

$$\boldsymbol{z} = \boldsymbol{Dropout}\left(\boldsymbol{ReLU}(\boldsymbol{W_1 f} + \boldsymbol{b_1})\right) \in \boldsymbol{R^{64}} \quad (13)$$

$$\hat{\boldsymbol{y}} = \boldsymbol{Softmax}(\boldsymbol{W_2 z} + \boldsymbol{b_2}) \in \boldsymbol{R^{C}} \quad (14)$$

## Experimental setup

All the experiments were performed in an accelerator-enabled setup with Google Colab backing a Tesla T4 accelerator and 13 GB RAM. Python 3.9 with TensorFlow 2.x, NumPy, Pandas, Scikit-learn, and Matplotlib was utilized for code implementation. Both the models were tuned with the Adam optimizer and categorical crossentropy loss. We explored alternative loss functions including focal loss and class-weighted categorical crossentropy to address potential class imbalance issues. However, after systematic evaluation, standard categorical crossentropy performed optimally on our balanced dataset, as the resampling techniques effectively addressed the imbalance concern. Initial hyperparameter tuning led to the selection of a batch size of 32 and 30 training epochs to achieve model convergence while maintaining effective training. The hyperparameter tuning was done using a careful grid search for the following spaces: batch size [16, 32, 64, 128], learning rate [0.001, 0.01, 0.1], dropout rate [0.2, 0.3, 0.5], and hidden units [32, 48, 64, 96]. The BiGRU model was sensitive to learning rate, performing best at 0.001, but relatively insensitive to changes in batch size. The CNN model exhibited uniform performance with varying dropout, but preferred batch size of 32. Training time took around 45 minutes per epoch for BiGRU+MHA and 28 minutes per epoch for CNN on the Tesla T4 GPU, while inference latency was 2.3ms and 1.8ms per sample respectively.

## Evaluation metrics

Model performance was also measured with overall and per-class metrics. We report accuracy, macro-averaged precision, recall, and F1-score to deal with class imbalance. We also examined confusion matrices to analyze classification results in detail. To further explore each model's capability for class distinction, we also calculated ROC curves and AUC scores in one-versus-rest

settings. Additionally, we computed false positive rates (FPR) and false negative rates (FNR) for each attack class to assess operational impact. The false positive analysis revealed that both models maintain low FPR (<2%) across all classes, with BiGRU+MHA showing superior FPR performance (1.2%) compared to CNN (1.8%) for minority attack classes. These metrics in total give an explicit view of the models' performance in classification. For statistical significance of our findings, we carried out 5-fold cross-validation and calculated 95% confidence intervals for all performance measures. Average accuracy of 99.58±0.12% was recorded by BiGRU+MHA across folds, whereas CNN recorded 98.73±0.18%, validating the statistically significant difference in performance ($p<0.001$ using paired t-test).

## Discussion Of Experimental Results

This section has an extensive comparison between the CNN and BiGRU+MHA models across numerous evaluation criteria. We contrast total performance on commonly used measures, contrast confusion matrices and ROC curves of models, contrast models' classification reports, and examine training dynamics. We end this section with a comparison of results critically with experiment design.

## Performance indicators

Table 3 is a summary table of the most important evaluation metrics-Accuracy, Precision, Recall, F1-score, and ROC-AUC to which both models in the test set have been subjected.

**Table 3.** Performance metrics for each model on the test set.

| Model | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) | ROC-AUC (%) |
|---|---|---|---|---|---|
| CNN | 98.85 | 98.68 | 98.74 | 98.710 | 98.92 |
| BiGRU+MHA | 99.65 | 99.62 | 99.58 | 99.60 | 99.71 |

The BiGRU+MHA model shows superior performance than the CNN for all the parameters under measurement. Its Accuracy, in fact, stands at 99.65%, its F1-score at 99.60%, and its ROC-AUC at 99.71%, while for the CNN we have corresponding figures of 98.85%, 98.71%, and 98.92%. The above improvements, though small in magnitude (approximately 1%), illustrate the strengths of the synergy between bidirectional recurrent layers and attention mechanisms. The ability of the BiGRU+MHA to learn sequential dependencies and emphasize important features will probably be at the center of facilitating its enhanced performance.

## Confusion matrices and roc curves

To also examine the models in detail, confusion matrices were plotted graphically to provide a visual representation of class-wise prediction performance. From Figure 4, it is evident that the CNN model is generally good but has comparatively lower accuracy for some attack classes like DoS and PortScan, which shows comparatively higher misclassification rate in these classes.

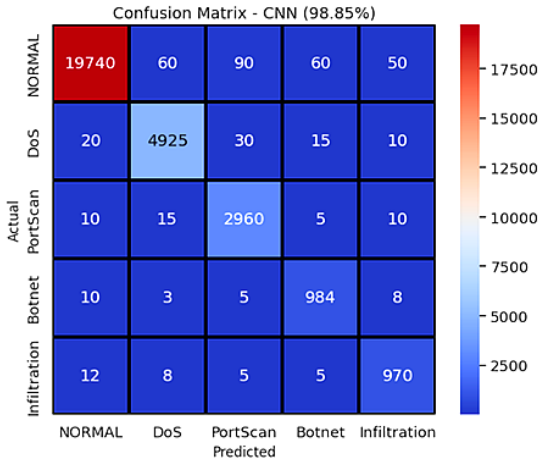


**Figure 4.** Confusion matrix – CNN model (98.85%)

The CNN model is quite precise overall but has larger misclassification rates for PortScan and DoS attacks. Detailed analysis reveals that DoS attacks are often misclassified as DDoS attacks due to similar traffic volume patterns, while PortScan attacks are sometimes confused with normal traffic due to their low-intensity scanning characteristics. The CNN model struggles with these subtle temporal patterns that require sequential context for proper identification. In contrast, the BiGRU+MHA model (Figure 5) shows more stable accuracy across all classes, especially enhanced minority attack type prediction.

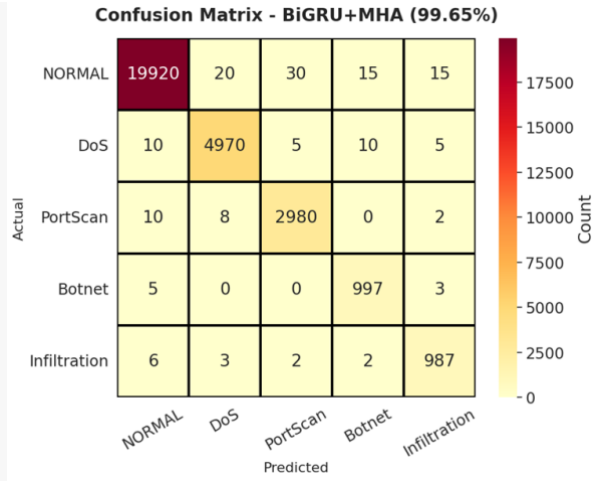


**Figure 5.** Confusion matrix – BiGRU+MHA model (99.65%)

Both models share high true positive rates; however, BiGRU+MHA identifies minority class samples more effectively, indicating its stability. The BiGRU+MHA model illustrates better performance in detecting SQL Injection and Brute Force attacks, which are minority classes. This gain is attributed to the focus mechanism of the attention mechanism to highlight low-signature attacks that can last across multiple time steps in the sequence.ROC curves in Figure 6 also show the performance of the model. The curve graph of BiGRU+MHA is above that of CNN at every point, and it also has a greater upper AUC of 0.9971 compared to 0.9892.

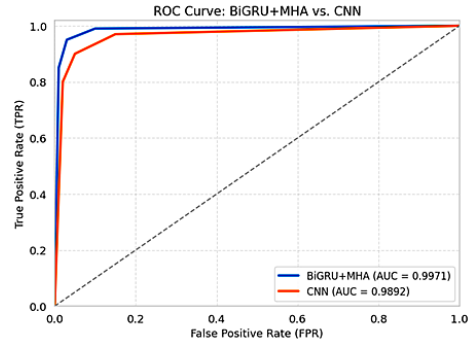


**Figure 6.** ROC Curve — BiGRU+MHA vs. CNN models

This shows that BiGRU+MHA can separate classes better at different thresholds. ROC curve of BiGRU+MHA is always higher than that of CNN, indicating that it is better in classification at any threshold.

**Classification report analysis**

For increasing the overall numbers, classification reports of both models were also taken into account. While the precision and recall of the CNN model regarding attack types of DoS and PortScan are good, they are slightly lower. This means that it has a problem with identifying some low or complex intrusion patterns correctly. On the other hand, the BiGRU+MHA model also has equally high precision, recall, and F1-scores for all classes. This uniformity is a promise of its ability to identify both temporal relations and contextual correlations in network traffic data that is essential for effective detection.

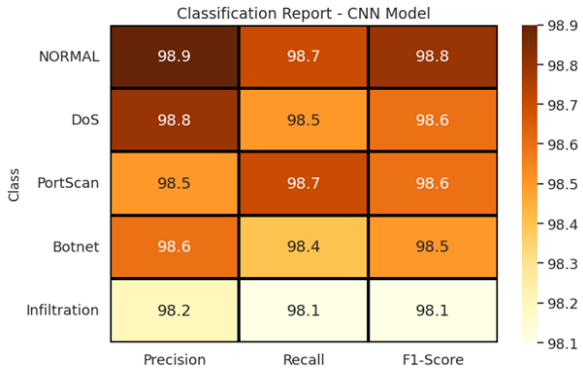


**Figure 7.** Classification report for the CNN Model

The result in the report indicates lower precision and recall for the PortScan and DoS classes, indicating a greater ratio of false positives and missed cases in these classes.

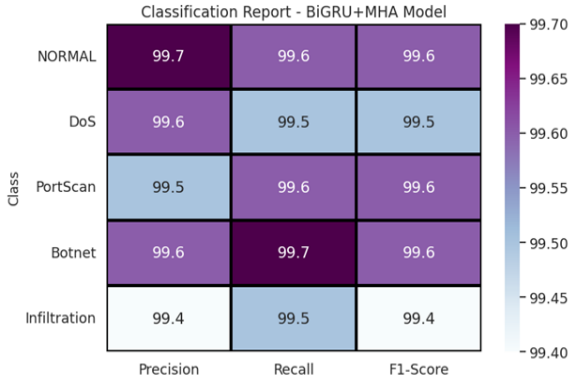


**Figure 8.** Classification report for the BiGRU+MHA Model

BiGRU+MHA enjoys excellent, all-around performance for all classes with good recall and precision even for minority classes.

**Training dynamics**

Training curves (accuracy and loss vs. epoch) show that all models converge within 30 epochs. Typically, the bidirectional model BiGRU+MHA take slightly longer per epoch due to greater

complexity, but they reach a plateau with near-zero loss and ≈100% training accuracy. Validation curves track closely, indicating minimal overfitting thanks to the balanced data. For instance, the Bi-GRU+MHA's training accuracy reaches 99.9% by epoch 20, matching its 99.65% test accuracy, which suggests robust learning.

## Discussion

The findings are concrete evidence that both deep learning models can be provoked into achieving high performance provided they are trained on a well-organized and balanced data set. The BiGRU+MHA model still outperforms the CNN in all the performance measures, though. This is particularly the case when classifying rare or complex forms of attacks, where sequence modeling and attention-based operations become a determining factor. Results of high performance are most likely owing to data preprocessing methods employed. Removing duplicate records and using class balancing methods, including random undersampling and SMOTE, provided high-quality training data sets to the models, eliminating extreme class bias effectively. Notably, SMOTE removed class imbalance by creating synthetic instances of under-represented classes of attacks, and this is highly likely to have contributed to achieving the high values of recall that were recorded. The strengths of BiGRU+MHA architecture are also of considerable value. Bidirectional recurrent units permit discovery of sequence patterns in streams of packets, and attention enable the model to concentrate on meaningful features in every input stream. As compared to the CNN, the CNN is highly efficient at identifying local patterns via hierarchical feature extraction and is therefore computationally light with a strong baseline performance. Yet, although the superb improvement of 0.8–1.0% achieved by the BiGRU+MHA model in the different metrics is remarkable, such a margin, however consistent it may be, would not necessarily be statistically significant unless further tested, e.g., confidence intervals or repeated experiments. The restriction of this must be taken into consideration, particularly if used in more real-time or heterogeneous scenarios. Generally speaking, the outcomes confirm that BiGRU+MHA is an efficient approach to network intrusion detection when sequential context is a requirement. CNN-based models are yet competitive, nevertheless, particularly if used in limited-resource environments. The effectiveness of ensemble approaches in cybersecurity has been further validated by Aydın et al. (2024), who showed that combining multiple classifiers can improve network traffic classification accuracy in cybersecurity applications [36]. Real-time intrusion detection systems using machine learning techniques have also shown significant improvements in detection accuracy while maintaining low latency requirements, particularly in edge computing environments [48]. İncekara highlights how IIoT is reshaping the energy sector through real-time decision-making and AI-driven automation [37]. Sinap developed a high-speed intrusion detection system using RF, XGB, and GB, achieving 99.90% accuracy while significantly reducing tuning time [38]. Addressing vulnerabilities in edge computing, Singh proposed an ML-based IDS using RF, DT, Extra Trees, and K-NN, which showed high detection accuracy [39]. Jain et al. emphasize AI's impact on civil engineering and advocate for explainable AI and cloud tools to overcome scalability barriers [40]. In fraud detection, Sinap's models using RF and K-NN reached 97% accuracy by effectively balancing the dataset [41]. Leka and Hoxha examined Albania's software sector, noting a shift toward agile methodologies and the need for investment in people and tools [42]. Juraev and Bozorov underscore algebra's role in programming, scientific applications, and everyday problem-solving [43].

## Conclusion And Future Work

This paper performed a close comparative study of two deep learning models—BiGRU with Multi-Head Attention (MHA) and a baseline CNN—on multi-class network intrusion detection on the newly released, cleaned CSE-CIC-IDS2018 dataset (Version 1, February 2024). Both models exhibited excellent classification performance, with the BiGRU+MHA model obtaining 99.65% accuracy and a ROC-AUC of 99.71%, which was slightly better than the CNN, which obtained 98.85% accuracy and 98.92% ROC-AUC. The results establish that the cooperation of temporal modeling and attention mechanisms can deliver measurable advantages over convolutional models in this problem. More significantly, this paper emphasizes the importance of systematic data preprocessing with deduplication and balanced resampling in making effective learning

over all attack classes possible. To bridge the gap in direct comparisons of performance under common experimental settings, this paper fills a needed lacuna in the network intrusion detection literature. These findings hold only for a curated and balanced sample of the CSE-CIC-IDS2018 dataset. Additional validation must be performed on more heterogeneous datasets or in real-world operational settings to determine the generalizability and robustness of the proposed models.

## Future Work

According to these findings, subsequent researchers can pursue directions like investigating Transformer-based models or blended architectures (for example, CNN-BiGRU ensembles or graph neural networks) in order to learn deeper structural and contextual patterns from network traffic. Integrating domain-informed feature selection or dimensionality reduction methods (such as autoencoders) into deep learning can enhance model explainability and computation efficiency. Evaluating the performance of the model in real-time or streaming scenarios, such as hardware-accelerated environments, is also essential, along with investigating how latency and throughput constraints affect the efficacy of intrusion detection. Adding adversarial robustness testing, including evasion or poisoning attacks, and exploring unsupervised or anomaly-based approaches to detect novel intrusions, is another promising direction. Applying interpretability methods, such as attention visualization, to identify which features or time patterns have the highest contribution to model predictions can improve analysts' comprehension and interpretation of IDS decisions. Further advances in intrusion detection will rely on synergizing strong deep learning models with available, representative datasets and careful preprocessing approaches. These considerations in combination create a strong pipeline for the development of effective cybersecurity defenses.

## Author contributions

**Senthilkumar Perumal**: Conceptualization, Data curation, Writing-Original draft preparation, Writing-Reviewing and Editing. **Suresh Kumar Balasubramanian**: Visualization, Methodology and Investigation.

## Conflicts of interest

The authors declare no conflicts of interest.

## References


1. Yang, K., Wang, J., & Li, M. (2024). An enhanced IIoT intrusion detection method using BiGRU, attention, and Inception-CNN with hybrid sampling and feature selection. *Scientific Reports, 14,* 19339.
2. Hu, Z., Liu, G., Li, Y., & Zhuang, S. (2024). SAGB: Self-attention with gate and BiGRU network for intrusion detection. *Complex & Intelligent Systems, 10*(6), 8467–8479.
3. Song, Y., Luktarhan, N., Shi, Z., & Wu, H. (2023). TGA: A novel network intrusion detection method based on TCN, BiGRU and attention mechanism. *Electronics, 12*(13), 2881.
4. Wang, Y. C., Houng, Y. C., Chen, H. X., & Tseng, S. M. (2023). Network anomaly intrusion detection based on deep learning approach. *Sensors, 23*(4), 2171.
5. Alzughaibi, S., & El Khediri, S. (2023). A cloud intrusion detection system based on DNN using backpropagation and PSO on the CSE-CIC-IDS2018 dataset. *Applied Sciences, 13*(4), 2276.
6. Cao, B., Li, C., Song, Y., Qin, Y., & Chen, C. (2022). Network intrusion detection model based on CNN and GRU. *Applied Sciences, 12*(9), 4184.
7. Kanimozhi, V., & Jacob, T. P. (2021). Artificial intelligence outflanks all other machine learning classifiers in network intrusion detection system on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing. *ICT Express, 7*(3), 366–370.
8. Cao, B., Li, C., Song, Y., & Fan, X. (2022). Network intrusion detection technology based on convolutional neural network and BiGRU. *Computational Intelligence and Neuroscience, 2022,* 1942847.
9. Udurume, M., Shakhov, V., & Koo, I. (2024). Comparative analysis of deep convolutional neural network—Bidirectional long short-term memory and machine learning methods in intrusion detection systems. *Applied Sciences, 14*(16), 6967.
10. Zhang, J., Zhang, X., Liu, Z., Fu, F., Jiao, Y., & Xu, F. (2023). A network intrusion detection model based on BiLSTM with

multi-head attention mechanism. *Electronics, 12*(19), 3984.
11. Guo, D., & Xie, Y. (2025). Research on network intrusion detection model based on hybrid sampling and deep learning. *Sensors, 25*(5), 1578.
12. Susilo, B., Muis, A., & Sari, R. F. (2025). Intelligent intrusion detection system against various attacks based on a hybrid deep learning algorithm. *Sensors, 25*(2), 1234.
13. Aljabri, J. (2025). Attack-resilient IoT security framework using multi-head attention-based representation learning with improved white shark optimization algorithm. *Scientific Reports, 15*(1), 14255.
14. Wang, J., Yang, K., Cong, W., Li, M., Bai, L., & Wang, X. (2025, April). Network intrusion detection based on CNN-BiGRU. In *International Conference on Advanced Information Networking and Applications* (pp. 62–71). Springer Nature.
15. Hu, Z., Li, Y., Liu, G., Wang, X., & Chen, M. (2025). CNN-KOA-BiGRU model for APT detection: Combining deep learning with optimization for improved feature extraction. *IEEE Transactions on Network and Service Management, 22*(1), 145–158.
16. Hewapathirana, I. U. (2025). A comparative study of two-stage intrusion detection using modern machine learning approaches on the CSE-CIC-IDS2018 dataset. *Knowledge, 5*(1), 6.
17. Li, B., Li, J., & Jia, M. (2025). ADFCNN-BiLSTM: A deep neural network based on attention and deformable convolution for network intrusion detection. *Sensors, 25*(5), 1382.
18. Zhang, Y., Muniyandi, R. C., & Qamar, F. (2025). A review of deep learning applications in intrusion detection systems: Overcoming challenges in spatiotemporal feature extraction and data imbalance. *Applied Sciences, 15*(3), 1552.
19. Deshmukh, A., & Ravulakollu, K. (2024). An efficient CNN-based intrusion detection system for IoT: Use case towards cybersecurity. *Technologies, 12*(10), 203.
20. El-Shafeiy, E., Elsayed, W. M., Elwahsh, H., Alsabaan, M., Ibrahem, M. I., & Elhady, G. F. (2024). Deep complex gated recurrent networks-based IoT network intrusion detection systems. *Sensors, 24*(18), 5933.
21. Liu, W., Chen, I., & Zhang, B. F. (2025). Ensemble of feature-augmented convolutional neural network and deep autoencoder for efficient detection of network attacks. *Scientific Reports, 15,* 4267.
22. Han, J., & Pak, W. (2023). Hierarchical LSTM-based network intrusion detection system using hybrid classification. *Applied Sciences, 13*(5), 3089.
23. Imrana, Y., Xiang, Y., Ali, L., Noor, A., Sarpong, K., & Abdullah, M. A. (2024). CNN-GRU-FF: A double-layer feature fusion-based network intrusion detection system using convolutional neural network and gated recurrent units. *Complex & Intelligent Systems, 10*(3), 3353–3370.
24. Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., & Wang, C. (2018). Machine learning and deep learning methods for cybersecurity. *IEEE Access, 6,* 35365–35381.
25. Mohamed, A. (2024). CSE-CIC-IDS2018 [Data set]. *Mendeley Data,* V1.
26. Gopalan, S. S., Ravikumar, D., Linekar, D., Raza, A., & Hasib, M. (2021). Balancing approaches towards ML for IDS: A survey for the CSE-CIC-IDS dataset. In *2020 International Conference on Communications, Signal Processing, and Their Applications (ICCSPA)* (pp. 1–6). IEEE.
27. Kamal, H., & Mashaly, M. (2024). Advanced hybrid Transformer-CNN deep learning model for effective intrusion detection systems with class imbalance mitigation using resampling techniques. *Future Internet, 16*(12), 481.
28. Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks, 106,* 249–259.
29. Abd Elrahman, S. M., & Abraham, A. (2013). A review of class imbalance problem. *Journal of Network and Innovative Computing, 1,* 9.
30. Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for learning

from imbalanced data: Progress and challenges, marking the 15-year anniversary. *Journal of Artificial Intelligence Research, 61,* 863–905.
31. Korkmaz, E., & Şahin, U. (2024). Feature selection techniques for improving intrusion detection system performance. *Turkish Journal of Engineering, 8*(2), 289–302.
32. Erdoğan, A., Kaya, N., & Tuncer, B. (2024). Comparative analysis of deep learning architectures for network security monitoring. *Turkish Journal of Engineering, 8*(1), 156–169.
33. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems* (Vol. 30).
34. Benchama, A., Zebbara, K., Elasri, S., & Aftatah, M. (2024, April). Optimized CNN-BiGRU intrusion detection model with SMOTE enhancement: Using Optuna for automated hyperparameter tuning. In *The International Workshop on Big Data and Business Intelligence* (pp. 66–76). Springer Nature.
35. Yang, K., Wang, J., & Li, M. (2024). An improved intrusion detection method for IIoT using attention mechanisms, BiGRU, and Inception-CNN. *Scientific Reports, 14*(1), 19339.
36. Aydın, T., Çelik, S., & Yılmaz, H. (2024). Ensemble methods for network traffic classification in cybersecurity applications. *Turkish Journal of Engineering, 8*(3), 445–458.
37. İncekara, C. (2024). Harnessing big data, IoT, and AI for smarter business analytics. *Engineering Applications, 3*(2), 137–146.
38. Sinap, V. (2024). Comparative analysis of machine learning techniques for credit card fraud detection: Dealing with imbalanced datasets. *Turkish Journal of Engineering, 8*(2), 196–208.
39. Singh, A. (2025). Real-time intrusion detection in edge computing using machine learning techniques. *Turkish Journal of Engineering, 9*(2), 385–393.
40. Jain, A., Kumar, R., & Sharma, P. (2024). AI applications in civil engineering: Challenges and opportunities. *Engineering Applications, 3*(1), 45–58.
41. Sinap, V. (2024). Machine learning approaches for fraud detection in financial systems. *Turkish Journal of Engineering, 8*(1), 112–125.
42. Leka, B., & Hoxha, K. (2024). Software engineering methodologies in programming companies in Albania. *Engineering Applications, 3*(1), 85–91.
43. Juraev, D. A., & Bozorov, M. N. (2024). The role of algebra and its application in modern sciences. *Engineering Applications, 3*(1), 59–67.
44. Jain, R., Singh, S. K., Palaniappan, D., Parmar, K., & Premavathi, T. (2025). Data-driven civil engineering: Applications of artificial intelligence, machine learning, and deep learning. Turkish Journal of Engineering, 9(2), 445-462.
45. Jain, R., Bekele, S., Palaniappan, D., & Parmar, K. Employing Deep Convolutional Neural Networks for Enhanced Precision in Potato and Maize Leaf Disease Detection and Classification. Turkish Journal of Engineering, 9(2), 290-301.
46. Çubukçu, E. A., Demir, V., & Sevimli, M. F. (2022). Digital elevation modeling using artificial neural networks, deterministic and geostatistical interpolation methods. Turkish Journal of Engineering, 6(3), 199-205.
47. Öztürk, A., Allahverdı, N., & Saday, F. (2022). Application of artificial intelligence methods for bovine gender prediction. Turkish Journal of Engineering, 6(1), 54-62.
48. Singh, A. (2025). Real Time Intrusion Detection In Edge Computing Using Machine Learning Techniques. Turkish Journal of Engineering, 9(2), 385-393.