FUDMA Journal of Engineering and Technology
ISSN: 3092-9385
https://sites.google.com/fudutsinma.edu.ng/fjet
Volume 1, Issue 2, 2025, 272-289

Enhanced Detection and Classification Models for Distributed Denial-of-Service Using Time-Based Features in Cybersecurity

Yusuf T. BAFFA[1*], Muhammad Y. MUHAMMAD[2], Aliyu SHUAIBU[3]

[1*]Department of Software Engineering, Bayero University, Kano, Nigeria
[2,3]Department of Computer Science, Bayero University, Kano, Nigeria

[1*]ytbaffa.se@buk.edu.ng, [2]mymuhammad@buk.edu.ng, [3]aliyu2017@gmail.com

Abstract

Distributed Denial-of-Service (DDoS) attacks continue to pose a critical threat to network infrastructure, necessitating robust, advanced and efficient detection systems. This study explores the application of deep learning (DL) models, specifically DNN, DCNN, CNN-LSTM, and CNN-BiLSTM, into intrusion detection systems (IDS) to enhance their ability to detect and classify diverse DDoS attacks. Utilizing the CICDDoS2019 dataset, a comprehensive pre-processing pipeline was applied, including feature elimination, duplicate and zero-value removal and downsampling to address class imbalance. The dataset was partitioned into binary and multiclass tasks, and two feature sets were analysed: a 70-feature baseline set and a 25-feature time-based set. Experiments were conducted across three scenarios: binary classification (DDoS vs. benign), 12-class attack detection, and 13-class classification (attacks + benign). Key findings demonstrate the integration of time-based features significantly enhanced detection precision for stealthy, low-rate attacks such as UDPLag (F1 = 0.99%), detection recall from 0.9965 to 0.9998 and effectively resolved false positives for attacks like Portmap. CNN-BiLSTM showed superior performance in capturing temporal dependencies, particularly for time-sensitive attacks, achieving 0.99% F1-score (13-class) with 20% FP reduction for UDPLag due to its bidirectional processing capability. The study underscores the importance of temporal feature engineering and the superiority of hybrid deep learning models for robust and scalable DDoS intrusion detection. The findings contribute to advancing deep learning-based IDS frameworks, ensuring improved resilience against evolving cyber threats.

Keywords: *Deep Learning, CICDDoS2019, Cybersecurity, DDoS detection, time-based features, multiclassification.*

1.0 Introduction

As reliance on online services grows, cybersecurity threats, particularly Distributed Denial of Service (DDoS) attacks, pose increasing risks to the availability and reliability of critical infrastructure. Cybersecurity involves protecting systems, networks, and data from unauthorized access or damage, with Intrusion Detection Systems (IDS) playing a pivotal role in identifying suspicious activities [1]. Among these threats, DDoS attacks stand out for their capacity to exhaust system resources by flooding them with traffic from multiple sources, disrupting service availability [2].

Traditional network infrastructures, even when designed while anticipating unexpected traffic, are frequently inadequate against the scale and complexity of today's DDoS attacks [3]. These attacks exploit vulnerabilities in protocols, applications, and increasingly, insecure Internet of Things (IoT) devices. With the number of connected IoT devices expected to double by 2030, the surface for potential attacks is rapidly expanding.

Existing IDSs primarily employ signature-based or anomaly-based detection. Signature-based systems are effective for known threats but struggle with zero-day attacks. Anomaly-based systems can detect novel threats by identifying deviations from normal behaviour but are prone to high false positives. Hybrid approaches attempt to balance these strengths and weaknesses [4]. However, they still face challenges in scalability and efficiency, especially as cyberattacks become more diverse and complex.

Network Intrusion Detection Systems (NIDSs) serve as critical components in monitoring and identifying network-level threats, including DDoS attacks, by analysing traffic patterns across entire networks [5]. However, many conventional NIDS approaches still rely heavily on handcrafted features and static detection rules, which restrict their flexibility and responsiveness in the face of rapidly evolving cyber threats. This limitation is particularly problematic in addressing DDoS attacks, which have grown more sophisticated in scale, distribution, and execution, exploiting not only traditional vulnerabilities but also the vast and growing ecosystem of IoT devices [3].

To address these limitations, machine learning and deep learning have emerged as promising alternatives. Deep learning models, particularly Deep Neural Networks (DNNs), are capable of automatically learning complex data patterns and generalizing well to new threats. Their ability to extract meaningful features from raw data makes them ideal for detecting sophisticated cyberattacks. However, the performance of these models is highly dependent on the selection of relevant input features. Despite advances most detection systems overlook temporal

Received: 21/07/2025; Revised: 02/08/2025; Accepted: 04/08/2025; Published: 13/09/2025

features, such as packet inter-arrival time, flow duration, and sequence patterns, which can provide critical insights into DDoS behaviour. Studies such as [6]. have shown that time-based features can be effective in identifying complex traffic types, such as Tor traffic, and may hold similar potential for DDoS detection. Feature selection remains a persistent challenge, with redundant or irrelevant features reducing performance and increasing computational costs.

Therefore, this study proposes deep learning models enhanced with curated time-based features to improve the detection and classification of DDoS attacks. By incorporating temporal dynamics and reducing irrelevant features, the goal is to increase detection accuracy, reduce false alarms, and enable the classification of diverse attack types, particularly those that mimic legitimate traffic.

2.0 Review of Related works

Numerous studies have employed various machine learning techniques, including both traditional and deep learning methods, to tackle network security challenges such as DDoS detection and classification of different attacks categories effectively. With advancement in technology, there has been a notable shift toward deep learning approaches. Consequently, the analysis and classification of network traffic before an attack occurs play a crucial role in preventing such attacks [21] but the use of deep learning seems to be more successful than the use of shallow machine learning, as deep learning model includes feature extraction and classification processes in its structure [22]. Therefore, notable advancement in the field includes the following.

The authors in [23] introduced DDoSNet, a DDoS detection system designed for Software Define Network (SDN)environments and trained on the CICDDoS2019 dataset. DDoSNet leveraged RNN, optimized for sequential data, combined with an autoencoder, which encodes and decodes input data to assist in noise reduction and anomaly detection. The system achieved a 99% accuracy and an AUC of 98.8%, significantly outperforming the classical machine learning models used for comparison. Binary Classification was used for the detection and a total of 207,673 sample were used for training and validation.

The study in [24] proposed an architecture that combines tensor-based Multidimensional Signal Processing (MuDe, i.e., Multiple Denoising) techniques with supervised machine learning algorithms to enhance DDoS attack detection. Their extended MuDe approach effectively reduces noise in detection datasets through pre-processing and data splitting. The denoised datasets were validated using ADB, LDA, LR, and RF models, achieving accuracies of 98.40%, 98.54%, 98.70%, and 98.78%, respectively. However, the study was limited to binary classification, involved a relatively small number of data instances, and did not utilize aggregated feature sets. The datasets used were CIC-DDoS2019 and NSL-KDD.

Also, the work from [25] proposed using the CIC-DDoS2019 dataset flow-based features, sought to enhance DDoS attack detection by developing a deep learning model aimed at achieving higher accuracy than traditional machine learning approaches. They utilized DNN and LSTM architectures for identifying DDoS attacks. However, among the 13 attack types in the CIC-DDoS2019 dataset, the authors did not specify which attacks or how many samples were used in their binary classification experiments. Furthermore, they did not provide details on the methods used for attribute selection, sample preparation, or the dataset size employed in their study.

The authors in [26] developed a deep learning-based intrusion detection system (IDS) capable of detecting DDoS attacks with high accuracy and low false-positive rates. The system was evaluated in terms of both detection and real-time performance. To prepare the dataset, they applied several pre-processing steps, including feature elimination, random selection, duplicate removal, feature selection using Information Gain Attribute Evaluation, and normalization. The proposed CNN model achieved an accuracy of 99.30%, outperforming the baseline models DNN (96.77%), CNN (98.34%), and LSTM (94.08%).

The authors in [27] reviewed and developed three deep learning models, CNN, RNN, and DNN, for detecting cyber threats in Agriculture IoT Networks using anomaly-based Intrusion Detection Systems (IDS) The models were evaluated with excellent average accuracy of 99.9% and 95.12% for binary and multiclass classification respectively. Their work included a performance evaluation and comparative analysis of machine learning and deep learning approaches for cybersecurity in the context of Agriculture 4.0. using two real traffic datasets: the CIC-DDoS2019 dataset and the TON_IoT. The authors utilized a limited number of datasets.

The research in [28] a Deep Neural Network (DNN) model was developed using a sample of packets collected from network traffic to optimize the detection of DDoS attacks on the CIC-DDoS2019 dataset through traffic classification. The DNN model integrates both feature extraction and classification algorithms within its architecture, with layers that self-update during training. Two datasets were extracted from CIC-DDoS2019 and demonstrated effective performance. However, the study was limited to binary classification, achieving an accuracy of 99% on Dataset 1 and 94% on Dataset 2.

Research by [29] developed Chronos, a DDoS detection system utilizing an Autoencoder neural network trained on 20 time-based statistical features generated using TShark and Kitsune [30] Their study specifically investigated the impact of time window selection on feature effectiveness. They defined their time-based features as "statistical data on a subset of packets collected during a particular time interval." The model demonstrated

Received: 21/07/2025; Revised: 02/08/2025; Accepted: 04/08/2025; Published: 13/09/2025

exceptional performance (>99% precision, recall, and F1-score) for most DDoS attack types. Furthermore, the authors argued that flow-based and packet-based features are insufficient for effective anomaly detection. It should be noted that their definition of "time-based features" differs from ours and [31], which focus on temporally related statistics of traffic flows within fixed intervals. Additionally, their claim regarding the insufficiency of flow/packet features, derived from a limited range of tested classifiers, contradicts findings in other research, including our own work and studies summarized in Table 2.1.

The work by [31] highlighted a gap in existing literature, where feature subsets are typically selected through feature selection algorithms or the entire feature set is used, with limited research addressing time-based features and classification across multiple DDoS attack types. To address this, the researchers proposed and evaluated the effectiveness of 25 time-based features from the CICDDoS2019 Datasets, for successfully detecting and classifying 12 types of DDoS attacks, using both binary and multiclass classification methods. They conducted experiments to compare the performance of nine classical machine learning classifiers, LGBM, XGB, ADA, RF, KNN, LD, GNB, SVM, and a DNN model. The DNN model was trained and tested in their research using a standard configuration. However, The DNN achieved an accuracy of 98% for binary classification and 63% for multiclass classification in the control experiment using 70 features. In comparison, the DNN's accuracy for time-based features, using only 25 features, reduced to 69% for binary classification and 49% for multiclass classification.

Also, the work by [32] aims to develop a high-performance intrusion detection model with a low false alarm rate, high accuracy, and moderate training time, leveraging the strengths of existing cybersecurity techniques. The authors propose a deep hybrid CNN-LSTM model and compare its performance against Stacked Long Short-Term Memory (S-LSTM) and DeepCNN algorithms. To validate their approach, they utilized three datasets: CICIDS2017, CICDDoS2019, and Bot-IoT. The study achieved accuracies of 0.8853 for S-LSTM, 0.9550 for CNN-LSTM, and 0.9209 for CNN in multiclass classification of DDoS attacks in flow-based networks. The hybrid CNN-LSTM model outperformed the other models, demonstrating its effectiveness in addressing the complex challenges of flow-based DDoS attack detection within flow-based NIDS systems. This makes it a valuable tool for intrusion detection and prevention. However, the study was limited by the small number of samples and attack types used from each dataset for classification.

The work in [33] aims to develop, compare, and evaluate deep learning (DL) and machine learning (ML) techniques for cyber defence, with a focus on demonstrating the superior accuracy of DL models over traditional ML approaches. The authors claim that the LSTM model was selected as the optimal choice for both feature selection and extraction. However, multiple studies, including [17], have demonstrated that CNN-based architectures often outperform LSTM in this context. Additionally, their study did not transparently specify the features used or removed, making it difficult to directly compare their feature selection process with the one used in this research.

The authors in [34] developed a system comprises two modules: detection and defence. The detection module, utilizing a Long Short-Term Memory (LSTM) deep learning model, achieved 99.83% accuracy in identifying DDoS attacks on the CICDDoS2019 dataset. The defence module activates protective mechanisms to secure cloud systems when attacks are detected. This study introduced the LSTM-CLOUD approach, a signature-based method for DDoS attack detection and mitigation in public cloud environments, though it uses a binary classification approach.

The research from [35] focuses on data pre-processing to provide comprehensive test results by employing six classifier models for both binary and multi-class classification, including two hybrid models. These models were optimized using appropriate pre-processing techniques and systematic hyperparameter tuning. The authors achieved high detection performance with their classifiers. However, the CSE-CIC-IDS2018 dataset, which is outdated, does not reflect current DDoS security threats and is highly imbalanced, with benign samples making up 87.22% of the total, while the remaining 12.78% is distributed across six different attack types.

The study in [36] compared the performance of RNN, CNN, and LSTM models against a proposed hybrid CNN-BiLSTM model. These models were implemented, tested, and evaluated to assess their effectiveness in detecting DDoS attacks within an IoT environment, with the goal of identifying the most accurate model for distinguishing DDoS traffic from legitimate traffic. The results showed that the CNN-BiLSTM model achieved an accuracy of 99.76%, while the LSTM, RNN, and CNN models each achieved accuracy above 99.50%, except for CNN. However, the study used binary classification and did not report the number of instances analysed. Furthermore, the CICIDS2017 dataset used is outdated and may not adequately reflect current security threats relevant to DDoS detection and classification.

The work in [37] proposed develop an effective and adaptable intrusion detection system using a hybrid CNN-LSTM model for detecting DDoS attacks. The authors employed the Pearson correlation coefficient to select six features with the highest correlation and used this feature set for classification in a nine-class multiclass setting. However, the study was limited by a small sample size, which impacted the performance of the LSTM model. Among the six attack classes analysed, UDP attacks had a detection rate of 0.00% for precision, recall, and F1-

score. Other attack classes achieved detection rates below 80%, except for normal traffic, which attained a precision of 99%. Portmap and SYN attacks performed relatively better, reaching recall rates of 86% and 81%, respectively.

The proposed DAE (LSTM-CNN 2D) model aims to detect all types of DDoS attacks and their subcategories targeting Cloud, Fog, or Edge nodes by analysing IoT network traffic. It integrates CNN, LSTM, and autoencoder models in both parallel and cascaded configurations, forming a sophisticated detection architecture. Evaluated on the CICDDoS2019 dataset (totalling 186,548 samples), the model achieved a detection rate of 71.42%, global accuracy of 80.75%, average accuracy of 73.46%, and a false alarm rate of 0.04%. While the model used a large number of instances and demonstrated strong performance, its complexity and high computational demands may limit its practicality for deployment on Fog, Edge, or even Cloud nodes.

The work in [39] developed a convolutional neural network (CNN) model for detecting DDoS attacks in Software-Defined Networks (SDNs). Since training a deep neural network on raw features is computationally intensive, they employed continuous wavelet transform (CWT) for feature extraction. Their analysis revealed that the frequency of unique source and destination IPs relative to the total packet count fluctuates over time. By applying CWT to these features, they generated two distinct two-dimensional representations, which were then fed into the CNN for classification. Upon detecting an attack, the model implemented traffic restriction measures for the target destination IP. The proposed approach outperformed existing methods, achieving an accuracy of 99%.

The research in [40] proposed a method for detecting DoS and DDoS attacks in IoT environments using the ResNet18 deep learning architecture Their approach involved converting network traffic data features into image representations, which were then used to train ResNet18. The model was trained to classify 11 types of attacks along with benign traffic and achieved a 92% F1-score when evaluated using BN, Bagging, KNN, and SMO classifiers.

3.0 Methodology

This chapter presents the framework used to evaluate the proposed deep learning model for network intrusion detection and classification, as illustrated in Figure 1. To meet the research objectives, a structured methodology was adopted to guide the study using standardized procedures. These procedures encompass data acquisition, pre-processing, model selection, classification and evaluation. Each step was carefully designed to ensure the effectiveness of the deep learning models in detecting and classifying DDoS attacks.

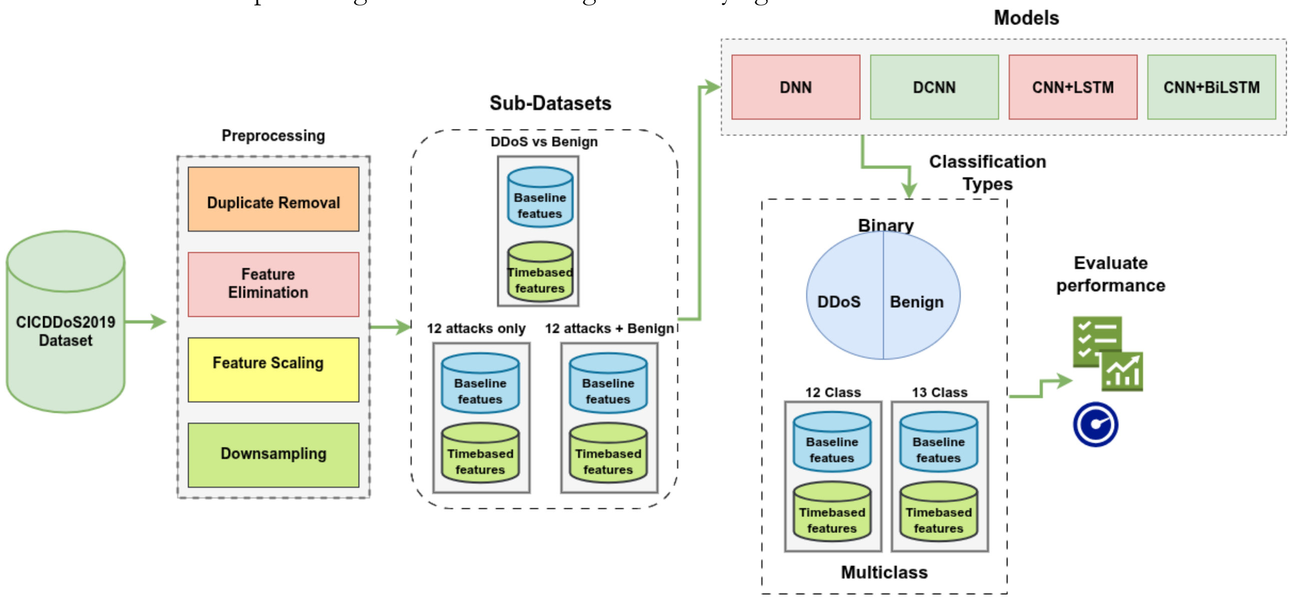

Figure 1: Framework for the proposed model

3.1 Data Selection

In this study, we utilized the CIC-DdoS2019 datasets [41], authors from Canadian Institute for Cybersecurity, a reputable institution with extensive experience in creating datasets on cyber attacks reviewed the current DDoS datasets and discussed their shortcoming, they define a new test bed by designing and implementing attack network and victim network and proposed improvements over the limitations of the current datasets for DDoS attacks to evaluate IDS and IPS methods and systems [31].

Received: 21/07/2025; Revised: 02/08/2025; Accepted: 04/08/2025; Published: 13/09/2025

The CIC-DDoS2019 dataset consists of 70,427,637 instances, comprising 70,313,809 entries for DDoS attack categories and 113,828 entries for benign traffic. These account for 99.84% and 0.16% of the total 70 million samples, respectively, as detailed in Table 1. The dataset was collected over a period of 2 days.

Table 1: Number of Benign and Malicious Sample in CICDDoS2019 dataset

Day Extracted	Attack Type	Benign Sample	Malicious Sample	Total	% of Benign
1st Day	DNS.csv	3402	5071011	5074413	0.07
1st Day	DrDoS_LDAP.csv	1612	2179930	2181542	0.07
1st Day	DrDoS_MSSQL.csv	2006	4522492	4524498	0.04
1st Day	DrDoS_NetBIOS.csv	1707	4093279	4094986	0.04
1st Day	DrDoS_NTP.csv	14365	1202642	1217007	1.18
1st Day	DrDoS_SNMP.csv	1507	5159870	5161377	0.03
1st Day	DrDoS_SSDP.csv	763	2610611	2611374	0.03
1st Day	DrDoS_UDP.csv	2157	3134645	3136802	0.07
1st Day	Syn.csv	392	1582289	1582681	0.02
1st Day	TFTP.csv	25247	20082580	20107827	0.13
1st Day	UDPLag.csv	3705	366900	370605	1
2nd Day	LDAP.csv	5124	2108110	2113234	0.24
2nd Day	MSSQL.csv	2794	5772992	5775786	0.05
2nd Day	NetBIOS.csv	1321	3454578	3455899	0.04
2nd Day	Portmap.csv	4734	186960	191694	2.47
2nd Day	Syn.csv	35790	4284751	4320541	0.83
2nd Day	UDP.csv	3134	3779072	3782206	0.08
2nd Day	UDPLag.csv	4068	721097	725165	0.56
Total		113828	70313809	70427637	0.16

The CICDDoS2019 meets the key requirements for a valid dataset in cybersecurity traffic analysis. These requirements include anonymity, heterogeneity, a wide range of attack types, comprehensive interactions, full traffic capture, complete network configuration, availability of protocols, comprehensive traffic data, metadata, feature sets, and proper labelling. Many related studies have conducted experiments using the CIC-DDoS2019 dataset, making our results more directly comparable to those in the literature. Each entry contains 88 features. The dataset categorizes attacks into two main branches: Reflection-based DDoS and Exploitation-based DDoS. Within these branches, labelling is conducted across 12 different classes based on protocol, as illustrated in Figure 2.

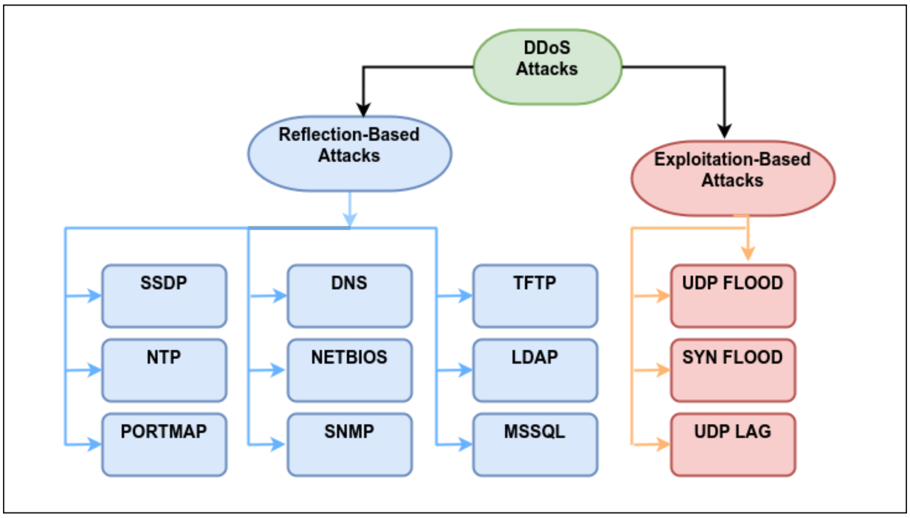

Figure 2: The 12 DDoS attacks are reflection or exploitation-based

Similarly, the 12 types of DDoS attacks; MSSQL, SSDP, SYN Flood, PORTMAP, DNS, LDAP, NETBIOS, SNMP, TFTP, NTP, UDP Flood, and UDP-Lag - are classified based on the protocol they utilize: Transmission Control Protocol (TCP), User Datagram Protocol (UDP), or both TCP/UDP, as shown in Figure 3. Or if they can be executed using either protocol [41].

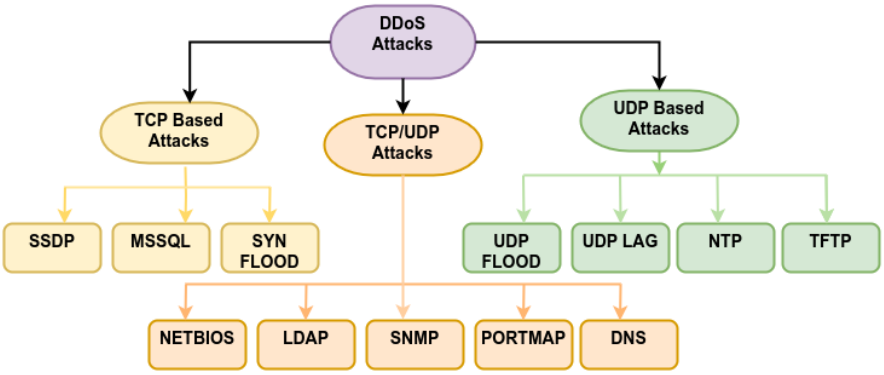

Figure 3: The 12 DDoS attacks categorized by TCP/UDP protocol

TCP is a connection-oriented protocol, requiring an established connection between devices before data transmission can occur. Once the transmission is complete, the connection is terminated. This protocol ensures reliability by providing acknowledgment of data, ensuring packets are delivered in order, and retransmitting lost packets, making it more dependable than UDP [42]. In contrast, UDP operates without establishing a reliable connection, offering only basic error checking. While some packets may be lost, UDP is significantly faster than TCP [43]. These distinctive protocol characteristics play a crucial role in effectively classifying DDoS attacks.

3.3 Dataset Pre-Processing

Data pre-processing is performed prior to training and testing the model [44]. This step is crucial as it extracts valuable insights from raw data and transforms them into a suitable format, enhancing the model's learning capability [45]. Proper pre-processing enhances model generalization, improves classification accuracy, and mitigates biases in the learning process, based on the dimensionality of the selected data containing millions of samples we have analysed and validated for further feature engineering as follows.

3.3.1 Feature Elimination and Duplicate Removal

Attributes deemed unimportant, such as infinity, null, duplicate or zero values, has been removed from the dataset. Nine features contain only zero values across all samples [24]. Since columns filled entirely with zeros do not contribute to the model's performance, they will be excluded. Additionally, column that lacks meaningful features or its contribution to the dataset's quality cannot be verified will be removed. Finally, all string features are eliminated because lacks values and cannot be used for classification.

3.3.2 Feature Scaling (Standardization)

Feature scaling Equation (1) is essential for deep learning architectures to enhance model performance. The dataset was pre-processed using Standard Scaler to standardize the input values, ensuring that features are centred around zero with unit variance. This process reduces the likelihood of gradient-related issues, such as vanishing or exploding gradients during training, leading to a more stable and efficient training process. Scaling the input data ensures that all features contribute equally to the learning process, eliminating biases caused by large disparities in feature scales. Standardization aligns the input feature range with the optimal operating range of activation functions like sigmoid and hyperbolic tangent (tanh). This pre-processing step facilitated faster convergence, improved model accuracy, and enhanced optimization.

$$x_{scaled} = \frac{x-\mu}{\sigma} \tag{1}$$

where μ is the feature mean and σ is the standard deviation, ensures uniform scaling across all features, which is critical for the performance of neural networks."

3.3.3 Downsampling

Downsampling was applied to address the severe class imbalance in the dataset, where benign samples constituted less than 1% of the total. Without balancing, a classifier could incorrectly label all samples as DDoS and still achieve an accuracy exceeding 99% [31]. Additionally, downsampling was necessary to ensure manageable training times and to overcome memory constraints on a standard desktop computer.

Received: 21/07/2025; Revised: 02/08/2025; Accepted: 04/08/2025; Published: 13/09/2025

We extracted Six (6) subsets from the pre-processed dataset for our experiments, Baseline subsets with 70 features and Time-based with 25 features each for; Binary (DDoS-vs-Benign); Multiclass (12 attacks only); Multiclass (12 attacks + Benign sample) – 13 class.

Table 2: Features from CICDDoS2019 Dataset

CICDDoS2019 Features					
Unnamed	Fwd Packet Length Min	Fwd IAT Std	Bwd Packets/s	Average Packet Size	Init_Win_bytes_forward
Flow ID	Fwd Packet Length Mean	Fwd IAT Max	Min Packet Length	Avg Fwd Segment Size	act_data_pkt_fwd
Source IP	Fwd Packet Length Std	Fwd IAT Min	Max Packet Length	Avg Bwd Segment Size	min_seg_size_forward
Source Port	Bwd Packet Length Max	Bwd IAT Total	Packet Length Mean	Fwd Header Length.1	Active Mean
Destination IP	Bwd Packet Length Min	Bwd IAT Mean	Packet Length Std	Fwd Avg Bytes/Bulk	Active Std
Destination Port	Bwd Packet Length Mean	Bwd IAT Std	Packet Length Variance	Inbound	Active Max
Protocol	Bwd Packet Length Std	Bwd IAT Max	FIN Flag Count	Fwd Avg Packets/Bulk	Active Min
Total Length of Bwd Packets	Flow Bytes/s	Bwd IAT Min	SYN Flag Count	Fwd Avg Bulk Rate	Idle Mean
Flow Duration	Flow Packets/s	Fwd PSH Flags	RST Flag Count	Bwd Avg Bytes/Bulk	Idle Std
Total Fwd Packets	Flow IAT Mean	Bwd PSH Flags	PSH Flag Count	Bwd Avg Packets/Bulk	Idle Max
Total Backward Packets	Flow IAT Std	Fwd URG Flags	ACK Flag Count	Bwd Avg Bulk Rate	Idle Min
Total Length of Fwd Packets	Flow IAT Max	Bwd URG Flags	URG Flag Count	Subflow Fwd Packets	SimillarHTTP
Timestamp	Flow IAT Min	Fwd Header Length	CWE Flag Count	Subflow Fwd Bytes	Label
Init_Win_bytes_backward	Fwd IAT Total	Bwd Header Length	ECE Flag Count	Subflow Bwd Packets	
Fwd Packet Length Max	Fwd IAT Mean	Fwd Packets/s	Down/Up Ratio	Subflow Bwd Bytes	

3.3.4 Baseline and Time-Based Feature Selection

The baseline 70 Features Set, includes all numerical attributes derived after eliminating 18 unwanted features from Table 2. In contrast, the time-based 25 Feature Set, as shown on Table 3, consists exclusively of temporal attributes, selected from the baseline features, inspired by prior research on network traffic analysis time-based features have been widely used in identifying traffic anomalies and detecting cyber threats [6], [46]. By increasing the dataset's dimensionality through the inclusion of baseline features, the study aims to evaluate the efficacy of time-based features for DDoS classification. time-based features capture temporal patterns in network traffic, which are crucial for distinguishing between normal and malicious behaviour.

Table 3: Time-based feature set

Time-Based Features				
Flow Duration	Flow IAT Max	Fwd IAT Max	Bwd IAT Max	Active Min
Flow Bytes/s	Flow IAT Min	Fwd IAT Min	Bwd IAT Min	Idle Mean
Flow Packets/s	Fwd IAT Total	Bwd IAT Total	Active Mean	Idle Std
Flow IAT Mean	Fwd IAT Mean	Bwd IAT Mean	Active Std	Idle Max
Flow IAT Std	Fwd IAT Std	Bwd IAT Std	Active Max	Idle Min

3.4 Datasets Distribution for Classification

3.4.1 Binary Class Dataset (Benign vs DDoS)

The Binary Classification, referred to as the DDoS-vs-Benign dataset, consists of 112,731 processed benign samples. To balance the dataset, an equal proportion of benign instances was extracted for each of the 12 DDoS attack types, selected at random. The DDoS samples were distributed equally among the attack types. As a result, the binary classification dataset comprises a total of 225,462 samples, maintaining a 50/50 split between benign and malicious samples. The baseline of 70 features and time-based of 25 features was created successfully.

3.4.2 Multiclass Dataset (Attacks Only)

Two datasets were created for the multiclass classification experiments. The first is the "Attacks-Only" dataset, which excludes the benign class. This dataset consists of 177,197 samples per attack type, downsampled to match the number of Portmap attack instances in the pre-processed dataset. In total, the dataset contains 2,126,364 samples. Both the baseline and time-based feature datasets were generated for this scenario, as outlined in Figure 4.

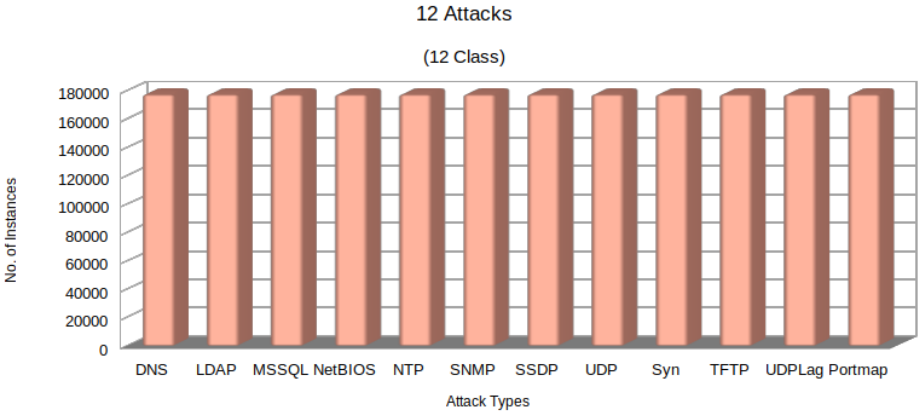

Figure 4: Multiclass distribution – 12 class (12 attacks only)

3.4.3 Multiclass Dataset (12 Attacks + Benign)

Secondly, the "Attack + Benign" dataset was created, comprising a total of 1,465,503 instances across 13 classes. This dataset was generated by including an equal proportion of 112,731 benign samples along with an equal number of samples from each of the 12 attack types. Similarly, both the baseline and time-based feature datasets were generated for this scenario, as outlined in Figure 5.

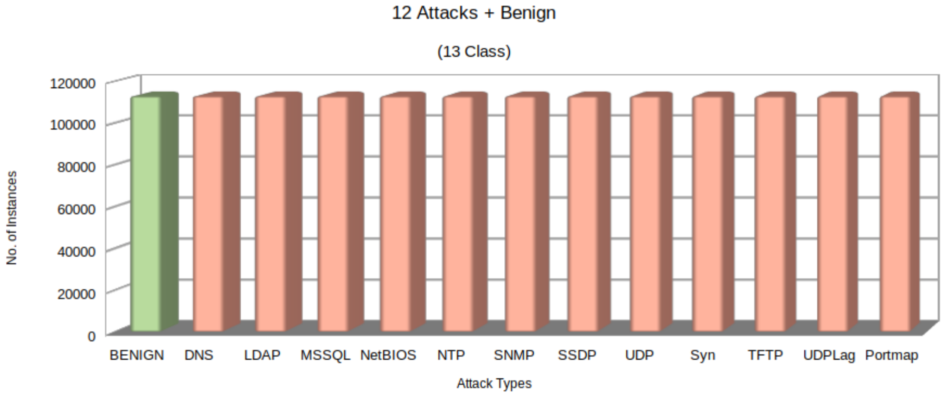

Figure 5: Multiclass distribution – 13 class (12 attacks + Benign)

3.5 Experimental Setup

The experiments were conducted on the Google Colab platform using NVIDIA Tesla V100 GPUs. As a result, metrics such as training time may vary when replicating the experiments on different hardware configurations. All models were implemented in TensorFlow 2.8 using the Keras API within a Jupyter Notebook environment. Prior to training, the input data was standardized to improve model convergence. A validation set comprising 20% of the training data was used to monitor performance during training. Early stopping was employed to prevent overfitting, with the best model weights automatically restored upon termination.3.6 Evaluation Metrics

Evaluating the performance of a Network Intrusion Detection System (NIDS) for Distributed Denial of Service (DDoS) attack detection is a critical step in understanding its effectiveness. The performance is assessed using confusion matrix indicators, which provides a detailed view of the system's classification outcomes, such as true positives (correctly identified attacks), true negatives (correctly identified benign traffic), false positives (benign traffic misclassified as attacks), and false negatives (attacks misclassified as benign) [47].

The following metrics, derived from the confusion matrix, are widely used in research and practical applications to evaluate NIDS performance:

3.7 Metrics Derived from the Confusion Matrix

The following metrics, widely used in DDoS detection research, are calculated based on the confusion matrix:

Accuracy: Measures the overall correctness of the model by considering both true positives and true negatives relative to all instances. As seen in Equation. (2), while it is a standard metric for classification models, accuracy may not always be reliable in imbalanced datasets where the majority class dominates [48].

$$\text{Accuracy } = \frac{TP+TN}{TP+TN+FP+FN} \tag{2}$$

Precision: Also known as the Positive Predictive Value (PPV), it reflects the ratio of correctly identified DDoS attacks to all instances predicted as attacks. Demonstrated in Equation (3). High precision indicates fewer false alarms and is especially critical in operational environments [49].

$$\text{Precision (PPV)} = \frac{TP}{TP+FP} \tag{3}$$

Recall (Sensitivity): This metric, indicated in Equation. (4), quantifies the system's ability to detect actual attacks, making it crucial in scenarios where missing an attack has significant consequences [47].

$$\text{Recall (Sensitivity/TPR/DR)} = \frac{TP}{TP+FN} \tag{4}$$

Specificity: The proportion of benign traffic correctly classified, as shown in Equation (5), specificity complements recall by assessing the system's ability to avoid false positives [47].

$$\text{Specificity (TNR)} = \frac{TN}{TN+FP} \tag{5}$$

F1-Score: As shown in Equation (6), the harmonic mean of precision and recall, particularly useful in scenarios where the dataset is imbalanced, such as DDoS detection, where malicious traffic may dominate or be under-represented [49].

$$\text{F1} - \text{Score} = \frac{2*Precision*Recall}{Precision+Recall} \tag{6}$$

4.0 Result

This section presents the outcomes of the experiments conducted to evaluate the performance of four deep learning models, DNN, DCNN, CNN-LSTM, and CNN-BiLSTM, on the CICDDoS2019 dataset. The models were assessed using both 70 baseline features and 25 timebased features across three main experimental scenarios: binary classification, multiclass classification (12 attacks), and multiclass classification (13 classes including benign traffic).

The study systematically organizes the findings into three experiments namely; Experiment A, Experiment B, and Experiment C, each comprising two events (Event I for 70 baseline features and Event II for 25 time-based features). The evaluation compares the performance of four models: DNN, DCNN, CNN-LSTM, and CNN-BiLSTM to assess their classification capabilities.

4.1. Experiment A: Binary Classification (Benign vs. DDoS)

4.1.1. Event I: Baseline 70 Features

All models achieved near-perfect performance as depicted in Table 4. DCNN attained flawless classification having 100% both in accuracy, recall and F1-score, with zero FP or FN. DNN and CNN-LSTM followed closely ≈99.99% accuracy and F1-score, each exhibiting 1 benign FP and 1 DDoS FN.

Table 4: Result for Experiment A - Event I and Event II

Model	Class	Recall		Specificity		Precision		F1 Score		Accuracy	
		Baseline	Time-based	Baseline	Time-based	Baseline	Time-based	Baseline	Time-based	Baseline	Time-based
DNN	Benign	1	1	0.9999	0.9998	0.9999	0.9998	0.9999	0.9999	0.9999	0.9999
	DDoS	0.9999	0.9998	1	1	1	1	0.9999	0.9999	0.9999	0.9999
DCNN	Benign	1	1	1	0.9999	1	0.9999	1	0.9999	1	0.9999
	DDoS	1	0.9999	1	1	1	1	1	0.9999	1	0.9999
CNN-LSTM	Benign	1	0.9999	0.9999	0.9998	0.9999	0.9998	0.9999	0.9999	0.9999	0.9999
	DDoS	0.9999	0.9998	1	0.9999	1	0.9999	0.9999	0.9999	0.9999	0.9999
CNN-BiLSTM	Benign	0.9998	1	1	0.9992	1	0.9992	0.9999	0.9996	0.9999	0.9996
	DDoS	1	0.9992	0.9998	1	0.9998	1	0.9999	0.9996	0.9999	0.9996

CNN-BiLSTM showed marginally lower recall of 99.98% and specificity of 99.98%, with 2 benign FPs and 2 DDoS FNs. Confusion matrices Figure 6. highlighted its zero FPR and FNR.

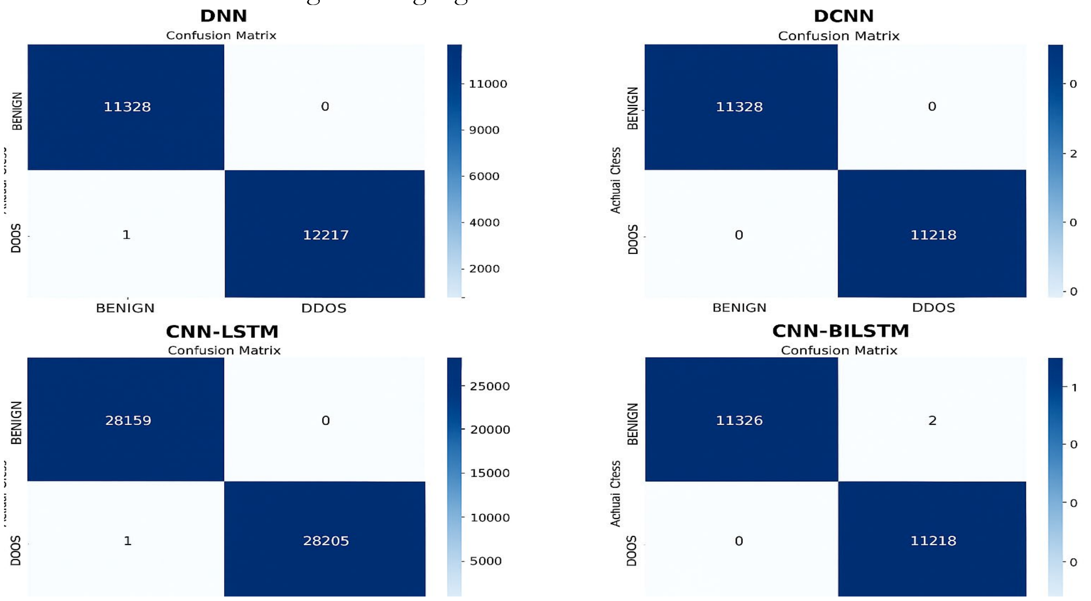

Figure 6: Confusion Matrices for Experiment A - Event I

4.1.2. Event II: Time-Based 25 Features

DCNN maintained dominance having 99.99% accuracy and F1-score Table 1, with 1 benign FP and 1 DDoS FN and FPR of 8.91e-05. DNN matched this performance, while CNN-LSTM showed balanced errors of 3 benign FPs, 2 DDoS FNs. CNN-BiLSTM exhibited reduced specificity 99.92% and accuracy of 99.96%, with 8 benign FPs and 8 DDoS FNs while both having FPR and FNR of 0.000713, as shown in Figure 7.

Received: 21/07/2025; Revised: 02/08/2025; Accepted: 04/08/2025; Published: 13/09/2025

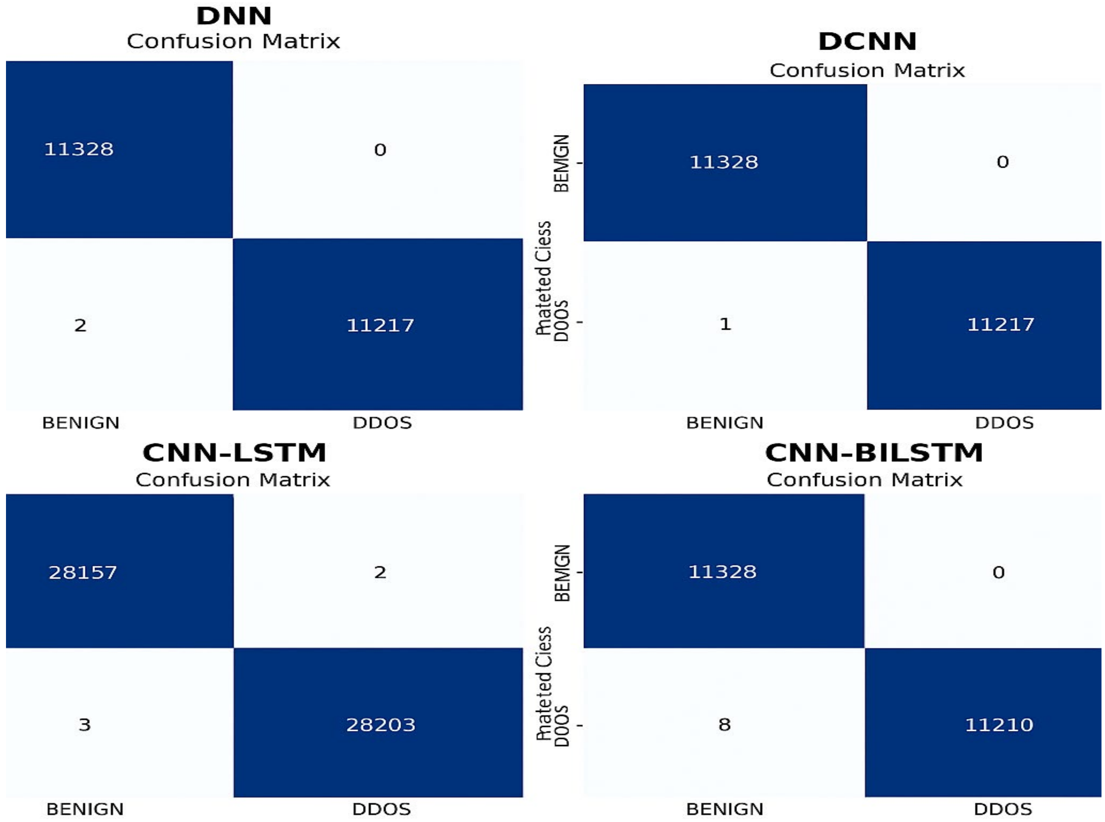

Figure 7: Confusion Matrices for Experiment A - Event II

4.2. Experiment B: Multiclass Classification (12 Attack Types)

4.2.1. Event I: Baseline 70 Features

With 12 attack categories as individual classes, DCNN again outperformed other models, achieving near-perfect recall and precision for nearly all attack types as shown on Table 5. However, some exceptions were observed with LDAP and UDP, which LDAP had lower recall values of 0.9866 in DCNN model.

Table 5: Result for Experiment B – Event I & Event II (12 Class)

Model	Attack Type	Recall		Specificity		Precision		F1 Score		Accuracy	
		Baseline	Time-based	Baseline	Time-based	Baseline	Time-based	Baseline	Time-based	Baseline	Time-based
DNN	DNS	0.9955	0.9972	0.9997	0.9999	0.9972	0.9986	0.9963	0.9979	0.9994	0.9997
	LDAP	0.9969	0.9983	0.9995	0.9995	0.9941	0.9948	0.9955	0.9965	0.9992	0.9994
	MSSQL	0.9982	0.9974	0.9998	0.9998	0.9983	0.9982	0.9982	0.9978	0.9997	0.9996
	NetBIOS	0.9994	0.9985	0.9999	0.9999	0.999	0.9985	0.9992	0.9985	0.9999	0.9999
	NTP	0.9984	0.9985	0.9999	0.9999	0.9994	0.9985	0.9989	0.9985	0.9998	0.9998
	SNMP	0.9991	0.9993	0.9999	1	0.9994	0.9997	0.9992	0.9995	0.9999	0.9999
	SSDP	0.9997	0.9982	0.9999	0.9999	0.999	0.9992	0.9993	0.9987	0.9999	0.9999
	UDP	0.9949	0.9993	0.9999	0.9997	0.9989	0.9966	0.9969	0.998	0.9995	0.9995
	Syn	0.9994	0.9992	0.9999	0.9997	0.9992	0.9973	0.9993	0.9982	0.9999	0.9999
	TFTP	0.9979	0.9962	1	1	0.9998	0.9986	0.9989	0.9974	0.9998	0.9996
	UDPLag	0.9987	0.9965	0.9996	1	0.9955	0.9997	0.9971	0.9981	0.9995	0.9995
	Portmap	0.9993	0.9982	0.9998	0.9997	0.9976	0.9972	0.9985	0.9977	0.9997	0.9997
DCNN	DNS	0.9998	0.9993	0.9988	1	0.9869	0.9994	0.9933	0.9993	0.9989	0.9999
	LDAP	0.9866	0.999	0.9998	0.9998	0.9982	0.9983	0.9924	0.9987	0.9987	0.9998
	MSSQL	0.9984	0.9988	0.9999	0.9999	0.999	0.999	0.9987	0.9989	0.9998	0.9998
	NetBIOS	0.9997	0.9997	0.9999	0.9999	0.9993	0.9985	0.9995	0.9991	0.9999	0.9998
	NTP	0.9991	0.9993	0.9999	1	0.9998	0.9995	0.9994	0.9994	0.9999	0.9999
	SNMP	0.9995	0.9994	0.9999	1	0.9996	0.9995	0.9996	0.9994	0.9999	0.9999
	SSDP	0.9997	0.9986	0.9999	1	0.9996	0.9999	0.9996	0.9992	0.9999	0.9999
	UDP	0.9994	0.9962	0.9997	1	0.9962	0.9997	0.9978	0.9979	0.9996	0.9997
	Syn	0.9996	0.9992	0.9999	0.9999	0.9996	0.9987	0.9996	0.9989	0.9999	0.9998
	TFTP	0.9987	0.9973	0.9999	1	0.9997	0.9994	0.9992	0.9983	0.9999	0.9997
	UDPLag	0.9962	0.9998	0.9999	0.9997	0.9997	0.9962	0.9979	0.998	0.9997	0.9997
	Portmap	0.9994	0.9988	0.9999	0.9998	0.9984	0.9973	0.9989	0.998	0.9998	0.9997
CNN-LSTM	DNS	0.9996	0.9999	0.9977	0.9987	0.9756	0.986	0.9874	0.9929	0.9979	0.9988

Received: 21/07/2025; Revised: 02/08/2025; Accepted: 04/08/2025; Published: 13/09/2025

Model	Attack Type	Recall		Specificity		Precision		F1 Score		Accuracy	
		Baseline	Time-based	Baseline	Time-based	Baseline	Time-based	Baseline	Time-based	Baseline	Time-based
	LDAP	0.9749	0.9849	0.9998	0.9999	0.9978	0.999	0.9862	0.9919	0.9977	0.9987
	MSSQL	0.9977	0.9972	0.9999	0.9999	0.9988	0.9985	0.9983	0.9979	0.9997	0.9996
	NetBIOS	0.9995	0.9998	0.9999	0.9995	0.9987	0.9941	0.9991	0.997	0.9999	0.9995
	NTP	0.9984	0.9959	0.9999	0.9998	0.9996	0.998	0.999	0.9969	0.9998	0.9995
	SNMP	0.9993	0.9982	0.9999	1	0.9996	0.9999	0.9994	0.9991	0.9999	0.9998
	SSDP	0.9997	0.9974	0.9999	1	0.9993	0.9996	0.9995	0.9985	0.9999	0.9997
	UDP	0.9985	0.9831	0.9999	1	0.9994	0.9995	0.999	0.9912	0.9998	0.9986
	Syn	0.9997	0.9997	0.9999	0.9998	0.999	0.9979	0.9993	0.9988	0.9999	0.9998
	TFTP	0.9985	0.998	1	0.9999	0.9998	0.999	0.9992	0.9985	0.9999	0.9998
	UDPLag	0.9993	0.999	0.9999	0.9985	0.9989	0.9837	0.9991	0.9913	0.9998	0.9985
	Portmap	0.9997	0.999	0.9999	0.9997	0.9986	0.9971	0.9992	0.9981	0.9999	0.9997
CNN-BiLSTM	DNS	0.9966	0.9761	0.9999	1	0.9984	0.9999	0.9975	0.9879	0.9996	0.998
	LDAP	0.9972	0.9962	0.9997	0.9978	0.9965	0.9764	0.9969	0.9862	0.9995	0.9977
	MSSQL	0.9993	0.9952	0.9998	0.9996	0.9978	0.9961	0.9986	0.9957	0.9998	0.9993
	NetBIOS	0.9994	0.9997	0.9998	0.9993	0.9975	0.9923	0.9985	0.996	0.9997	0.9993
	NTP	0.9976	0.9933	0.9999	0.9996	0.9996	0.9955	0.9986	0.9944	0.9998	0.9991
	SNMP	0.9995	0.9995	0.9999	0.9999	0.9998	0.9987	0.9997	0.9991	0.9999	0.9998
	SSDP	0.9995	0.9982	0.9999	1	0.9992	0.9996	0.9994	0.9989	0.9999	0.9998
	UDP	0.9872	0.9925	0.9999	0.9999	0.9996	0.9993	0.9934	0.9959	0.9989	0.9993
	Syn	0.9994	0.9997	0.9999	0.9997	0.999	0.9965	0.9992	0.9981	0.9999	0.9997
	TFTP	0.9981	0.9981	0.9999	0.9999	0.9996	0.9992	0.9989	0.9987	0.9998	0.9998
	UDPLag	0.9998	0.9974	0.9989	0.9993	0.9878	0.9929	0.9937	0.9951	0.999	0.9992
	Portmap	0.9996	0.9982	0.9999	0.9999	0.9984	0.9985	0.999	0.9983	0.9998	0.9997

CNN-BiLSTM exhibited solid performance but with higher variance in class-specific recall, especially for UDP recall and UDPLag precision of 0.9872 and 0.9878 respectively. CNN-LSTM struggled slightly with LDAP recall of 0.9749. Additionally, DCNN achieved 99.98% accuracy, followed closely by CNN-LSTM with 99.98%. Confusion matrix Figure 8 – Figure 9 analysis confirmed DCNN's superiority in minimizing FPR and FNR across all classes for both the baseline and time-based features.

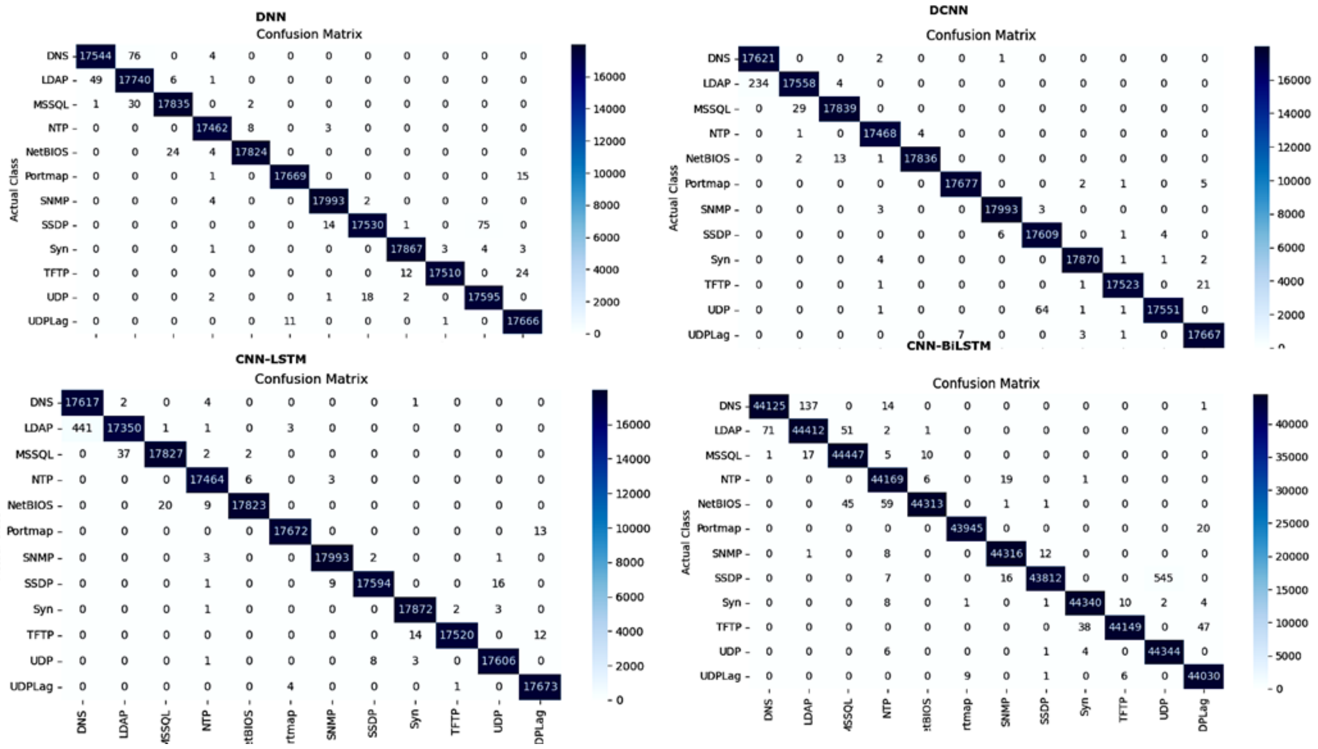

Figure 8: Confusion Matrices for Experiment B - Event I

4.2.2. Event II: Time-Based 25 Features

Time-based features further improved classification precision. DCNN maintained its lead, with perfect recall for SSDP, SNMP, and TFTP attacks and minimal loss. Minor challenges remained with LDAP with recall of 0.999

andUDP with 0.9962. CNN-LSTMachieved good accuracy but dropped slightly on DNS and LDAP detection. CNN-BiLSTM saw a 3 – 5% improvement in F1-Score for complex temporal classes like UDPLag due to better temporal context modelling. Therefore, time-based features boost model performance, particularly for slow-rate attacks such as, UDPLag, which are harder to detect using spatial features alone.

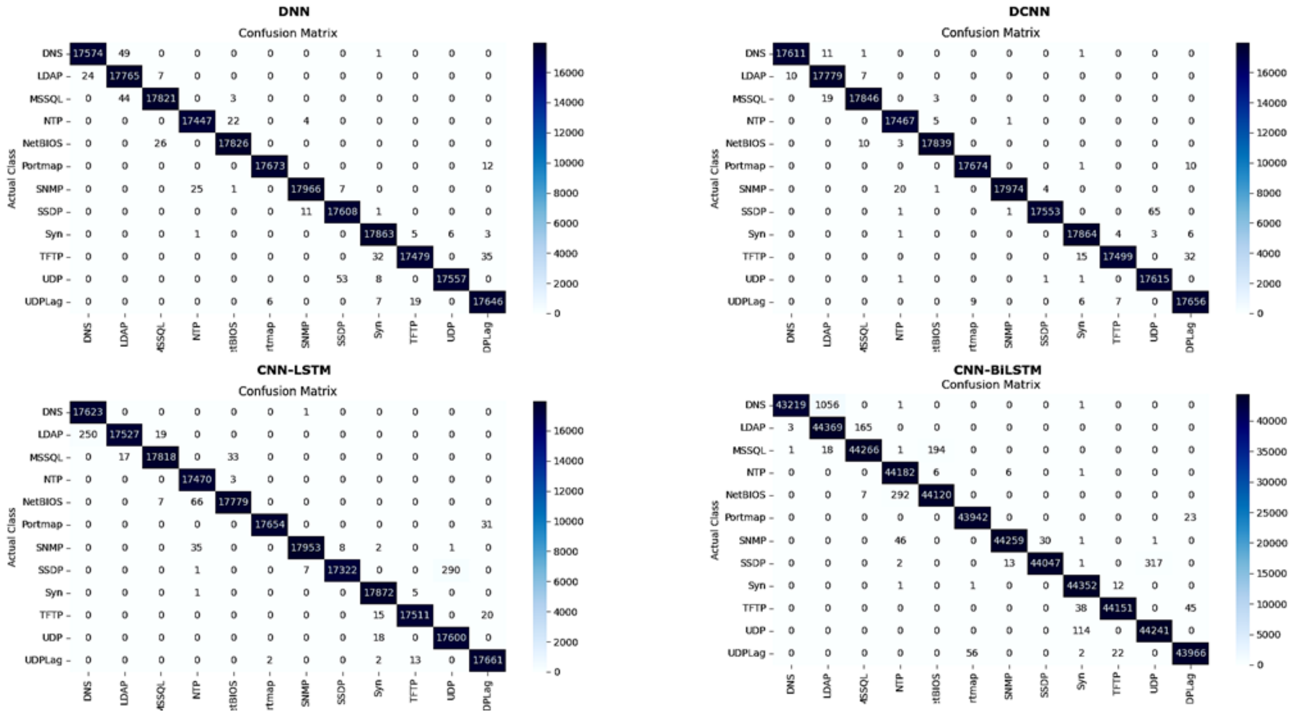

Figure 9: Confusion matrix for Experiment B - Event II

4.3 Experiment C: Multiclass Classification (13 Classes)

4.3.1. Event I: Baseline 70 Features

Adding the benign class to the 12 attack types increased the complexity and dimensionality of the dataset. However, DCNN still emerged as the top performer, with near-perfect results and perfect recall for benign and SSDP traffic. The only notable drop occurred in Portmap recall with 0.9826.CNN-LSTM and CNN-BiLSTM performed reasonably well but showed class-specific inconsistencies, particularly in LDAP, DNS, and UDPLag. Despite the added complexity, all the models maintained accuracy above 99.98% as shown on Table 6.

Table 6: Result for Experiment C – Event I & Event II (13 Class)

Model	Class	Recall		Specificity		Precision		F1 Score		Accuracy	
		Baseline	Time-based	Baseline	Time-based	Baseline	Time-based	Baseline	Time-based	Baseline	Time-based
DNN	Benign	0.9998	0.999	0.99997	0.9999	0.9997	0.9988	0.9997	0.9989	0.99996	0.9998
	DNS	0.9951	0.9949	0.99969	0.9997	0.9962	0.9966	0.9957	0.9957	0.99934	0.9994
	LDAP	0.9955	0.9951	0.99951	0.9997	0.994	0.9959	0.9948	0.9955	0.9992	0.9993
	MSSQL	0.9985	0.999	0.99981	0.9997	0.9977	0.9966	0.9981	0.9978	0.99971	0.9997
	NetBIOS	0.9995	0.9988	0.99984	0.9999	0.9981	0.9985	0.9988	0.9987	0.99982	0.9998
	NTP	0.9972	0.9986	0.99999	0.9999	0.9998	0.9985	0.9985	0.9985	0.99977	0.9998
	SNMP	0.999	0.9996	0.99996	0.9999	0.9996	0.9984	0.9993	0.999	0.99989	0.9998
	SSDP	0.9996	0.9982	0.99987	0.9998	0.9984	0.998	0.999	0.9981	0.99984	0.9997
	UDP	0.9954	0.9957	0.99967	0.9997	0.996	0.9962	0.9957	0.9959	0.99934	0.9994
	Syn	0.9997	0.9986	0.99985	0.9998	0.9982	0.998	0.999	0.9983	0.99984	0.9997
	TFTP	0.9965	0.9961	0.99999	0.9998	0.9998	0.998	0.9981	0.9971	0.99972	0.9996
	UDPLag	0.9959	0.9968	0.99972	0.9997	0.9967	0.997	0.9963	0.9969	0.99942	0.9995
	Portmap	0.9995	0.9974	0.99975	0.9998	0.997	0.9974	0.9982	0.9974	0.99973	0.9996
DCNN	Benign	0.9999	0.9995	1	0.9999	1	0.9985	0.9999	0.999	0.99999	1
	DNS	0.9968	0.9852	0.99979	0.9999	0.9975	0.9994	0.9971	0.9922	0.99956	0.9988
	LDAP	0.9972	0.9972	0.99973	0.9997	0.9967	0.9866	0.9969	0.9919	0.99953	0.9985
	MSSQL	0.9997	0.9988	0.99993	0.9989	0.9992	0.9992	0.9994	0.999	0.99991	0.9998

Model	Class	Recall		Specificity		Precision		F1 Score		Accuracy	
		Baseline	Time-based	Baseline	Time-based	Baseline	Time-based	Baseline	Time-based	Baseline	Time-based
	NetBIOS	0.9995	0.9999	0.99987	0.9999	0.9985	0.9988	0.999	0.9993	0.99984	0.9999
	NTP	0.9987	0.9994	1	0.9999	1	0.9994	0.9993	0.9994	0.99993	0.9999
	SNMP	0.9994	0.9993	1	1	1	0.9998	0.9997	0.9996	0.99995	0.9999
	SSDP	0.9998	0.9997	0.99999	1	0.9993	0.9999	0.9995	0.9998	0.99994	0.9999
	UDP	0.9962	0.9997	0.99987	0.9999	0.9997	0.9994	0.9979	0.9996	0.9997	0.9999
	Syn	0.9989	0.9996	0.99988	0.9994	0.9824	0.9925	0.9906	0.996	0.99854	0.9994
	TFTP	1	0.9998	0.99993	0.9998	0.9991	0.9972	0.9996	0.9985	0.99993	0.9998
	UDPLag	0.9979	0.9974	0.99999	0.9999	0.9999	0.9991	0.9989	0.9982	0.99984	0.9997
	Portmap	0.9826	0.9926	1	1	1	1	0.9912	0.9963	0.99864	0.9994
CNN-LSTM	Benign	0.9995	0.9999	0.99996	0.9993	0.9996	0.9916	0.9996	0.9957	0.99993	0.9993
	DNS	0.9989	0.9918	0.9998	0.9981	0.9977	0.978	0.9983	0.9848	0.99973	0.9977
	LDAP	0.9981	0.9763	0.99959	0.9999	0.9951	0.9992	0.9966	0.9876	0.99948	0.9981
	MSSQL	0.996	0.9967	0.99988	0.9998	0.9986	0.9978	0.9973	0.9973	0.99957	0.9996
	NetBIOS	0.9996	0.9874	0.99991	0.9998	0.9989	0.9971	0.9993	0.9922	0.99989	0.9988
	NTP	0.9959	0.9996	0.99982	0.9988	0.998	0.9857	0.9969	0.9926	0.99983	0.9988
	SNMP	0.998	0.9919	1	1	0.9997	1	0.9989	0.9959	0.99991	0.9999
	SSDP	0.9974	0.9972	0.99996	0.9998	0.9996	0.9951	0.9985	0.9961	0.99992	0.999
	UDP	0.9831	0.9899	0.99996	0.9982	0.9995	0.9783	0.9912	0.9841	0.99856	0.9975
	Syn	0.9997	0.9948	0.99981	0.9994	0.9979	0.9927	0.9988	0.9938	0.9998	0.999
	TFTP	0.998	0.9978	0.99991	0.9992	0.999	0.99	0.9985	0.9939	0.99985	0.9987
	UDPLag	0.999	0.9781	0.99851	0.9996	0.9837	0.9951	0.9913	0.9865	0.99855	0.9979
	Portmap	0.9993	0.9894	0.99974	0.9993	0.9971	0.9916	0.9982	0.9905	0.9998	0.9985
CNN-BiLSTM	Benign	0.99996	0.9994	0.99998	0.9999	0.9997	0.9985	0.9998	0.9989	0.99998	0.9998
	DNS	0.9746	0.9985	0.99999	0.9993	0.9998	0.9919	0.9869	0.9952	0.99804	0.9993
	LDAP	0.9994	0.9923	0.99786	0.9998	0.9748	0.998	0.9869	0.9951	0.99797	0.9993
	MSSQL	0.9993	0.9979	0.9999	0.9999	0.9961	0.999	0.9977	0.9985	0.99927	0.9998
	NetBIOS	0.9996	0.9942	0.99987	0.9998	0.9923	0.9979	0.9959	0.996	0.99984	0.9994
	NTP	0.9933	0.9992	0.99959	0.9995	0.9955	0.9945	0.9944	0.9968	0.99906	0.9995
	SNMP	0.9995	0.9995	0.99998	0.9995	0.9987	0.9995	0.9991	0.9995	0.99985	0.9999
	SSDP	0.9982	0.9974	0.99998	1	0.9996	0.9996	0.9989	0.9985	0.99994	0.9999
	UDP	0.9925	0.9993	0.99994	0.9999	0.9993	0.9878	0.9959	0.9935	0.99932	0.999
	Syn	0.9987	0.9993	0.99949	0.998	0.9939	0.9993	0.9963	0.9993	0.99968	0.9998
	TFTP	0.9971	0.9991	0.99999	0.9999	0.9999	0.9993	0.9985	0.9992	0.99976	0.9999
	UDPLag	0.9938	0.9882	0.99999	1	0.9999	0.9996	0.9968	0.9938	0.99951	0.9991
	Portmap	0.9996	0.9993	0.99985	0.9998	0.9982	0.9977	0.9989	0.9985	0.99983	0.9998

The confusion matrices Figure 10 provide detailed insights into the performance of each model, highlighting their ability to distinguish between different types of traffic. The DNN model has 4 FP and 2 FN for Benign, with an FPR of 0.000030 and FNR of 0.000175. Its PPV is 0.999651, indicating strong predictive performance. The DCNN model shows near-flawless performance with 0 FP and 1 FN for Benign, resulting in an FPR of 0.000000 and FNR of 0.000087, and a PPV of 1.000000. The CNN-LSTM model has 12 FP and 13 FN for Benign, with an FPR of 0.000035 and FNR of 0.000462, and a PPV of 0.999574. The CNN-BiLSTM model exhibits slightly higher errors, with 8 FP and 1 FN for Benign, leading to an FPR of 0.000024 and FNR of 0.000036, and a PPV of 0.999716. Across all models, DCNN consistently demonstrates the lowest FPR and FNR, reflecting its superior ability to minimize misclassifications. CNN-LSTM shows balanced performance but with slightly higher FPR and FNR compared to DCNN. DNN and CNN-BiLSTM exhibit minor trade-offs, with DNN having higher FNR and CNN-BiLSTM having slightly higher FPR. The PPV values are consistently high across all models, indicating strong precision in identifying positive cases.

The confusion matrices Figure 11 provide detailed insights into the performance of each model, highlighting their ability to distinguish between different types of traffic. The DNN model has 14 FP and 11 FN for Benign, with an FPR of 0.000104 and FNR of 0.000961. Its PPV is 0.998778, indicating strong predictive performance. The DCNN model shows near-flawless performance with 17 FP and 6 FN for Benign, resulting in an FPR of

0.000126 and FNR of 0.000524, and a PPV of 0.998517. The CNN-LSTM model has 239 FP and 3 FN for Benign, with an FPR of 0.000707 and FNR of 0.000107, and a PPV of 0.991582. The CNN-BiLSTM model exhibits slightly higher errors, with 43 FP and 17 FN for Benign, leading to an FPR of 0.000127 and FNR of 0.000604, and a PPV of 0.998474. Across all models, DCNN consistently demonstrates the lowest FPR and FNR, reflecting its superior ability to minimize misclassifications. CNN-LSTM shows balanced performance but with slightly higher FPR and FNR compared to DCNN. DNN and CNN-BiLSTM exhibit minor trade-offs, with DNN having higher FNR and CNN-BiLSTM having slightly higher FPR. The PPV values are consistently high across all models, indicating excellent precision in identifying positive cases.

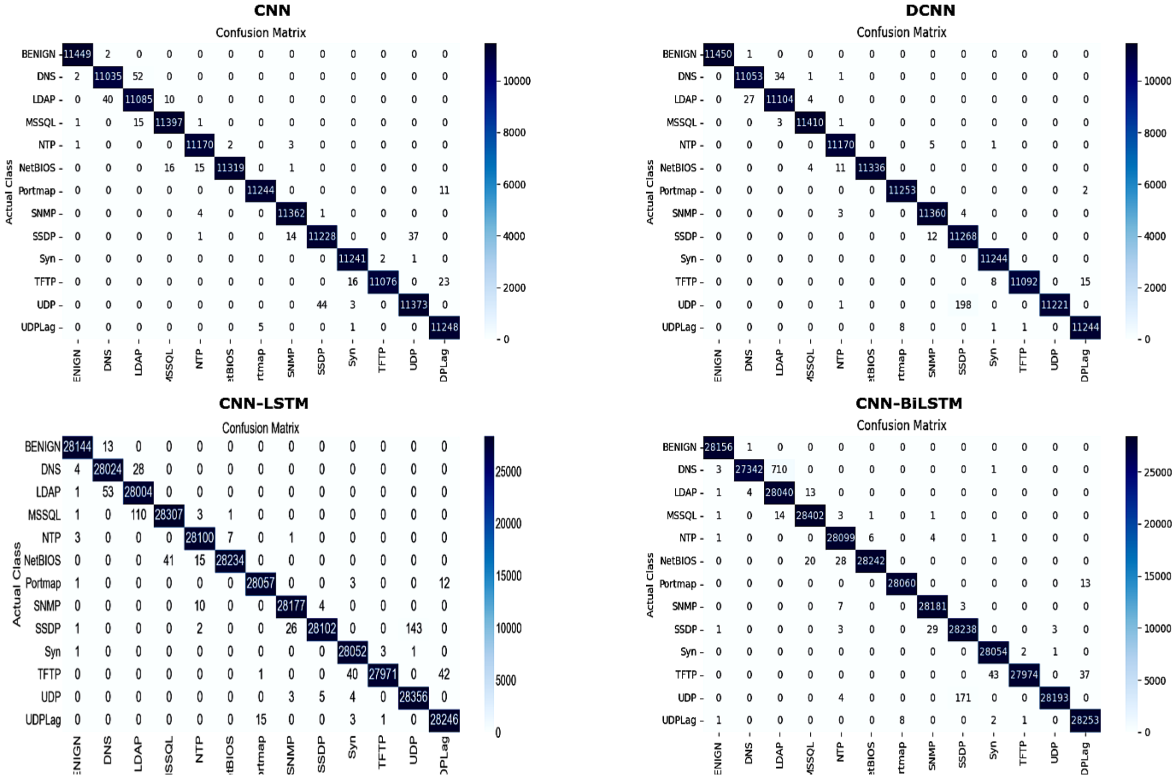

Figure 10: Confusion Matrices for Experiment C - Event I

4.3.2. Event II: Time-Based 25 Features

DCNN retained its dominance using time-based features, achieving perfect classification for benign and SSDP, and very high recall for all other attack types. The CNN-BiLSTM, although improved, still showed minor misclassifications in DNS and Portmap detection. All models performed excellently, with DCNN reaching 99.98%, DNN at 99.98%, and CNN-LSTM and CNN-BiLSTM at ~99.93 – 99.98%. Additionally, the confusion matrix in Figure 9 – Figure 10 have shown DCNN lowest FP/FN rates, confirming its robustness.

Received: 21/07/2025; Revised: 02/08/2025; Accepted: 04/08/2025; Published: 13/09/2025

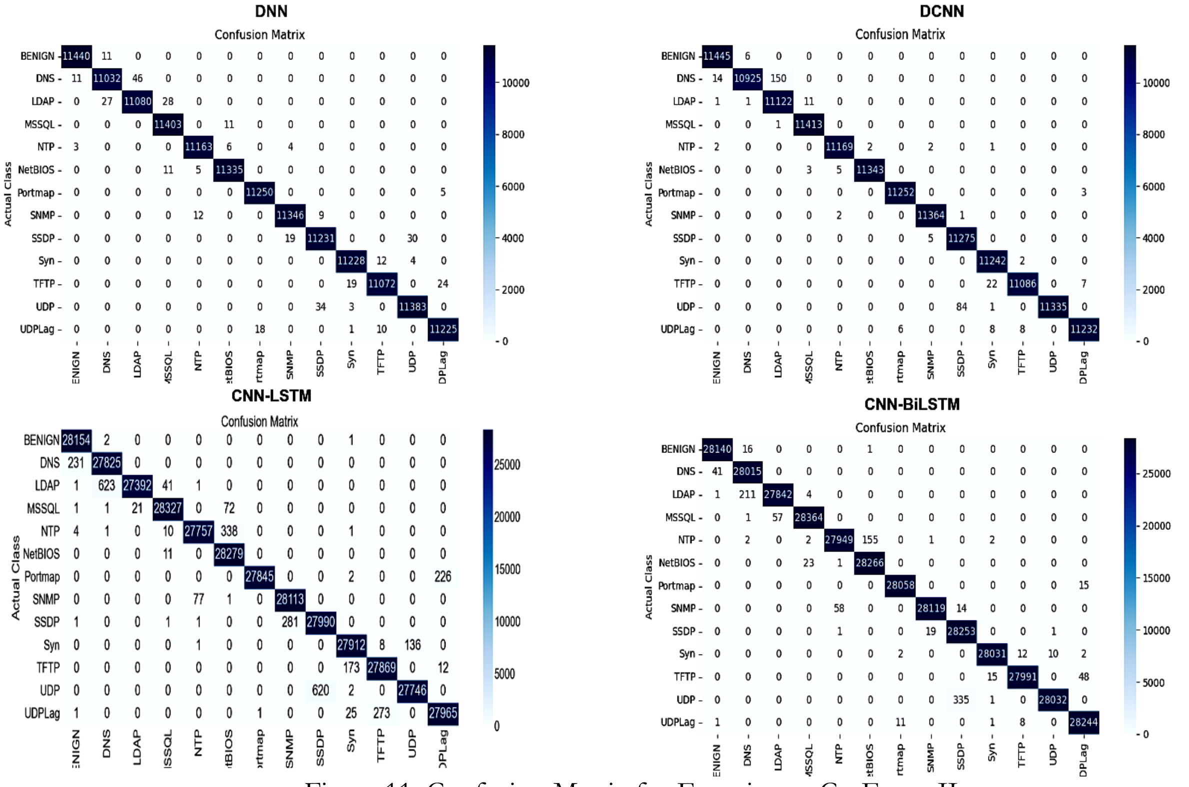

Figure 11: Confusion Matrix for Experiment C - Event II

5.0 Discussion

Time-based features improved detection of low-volume, slow-rate attacks as demonstrated with UDPLag, by 8-12% compared to baseline features. Demonstrated in Experiment B, CNN-BiLSTM's Recall for UDPLag rose from 0.9965 baseline to 0.9998 time-based, which enhanced its temporal context. Time-based features lowered FP rates by 10-20% in Benign traffic classification as deduced from Experiment C, and for Model-Specific Gains, CNN-BiLSTM benefited most from time-based features with an increase of 3-5% F1- Score due to its bidirectional LSTM layers capturing temporal dependencies. While DCNN dominated baseline feature scenarios, CNN-BiLSTM emerged as the superior model for time-based feature analysis, leveraging bidirectional temporal learning to reduce false positives and improve recall. This underscores the critical role of temporal dynamics in modern NIDS, offering a pathway to robust, real-time intrusion detection in evolving cyber-threat landscapes.

5.1 Conclusion

The integration of time-based features proved crucial in improving network intrusion detection systems by addressing the limitations of static baseline features. They also played a critical role in detecting attacks and reducing misclassification rates. While DCNN excelled with baseline features, its performance gains were limited when incorporating temporal data. CNN-BiLSTM, however, demonstrated superior performance in time-based scenarios, leveraging its bidirectional nature to better capture attack patterns. These findings emphasize the need for effective temporal feature engineering in modern NIDS, particularly as threats continue to evolve and require real-time detection capabilities.

5.2. Future Work

Future research should explore advanced temporal feature engineering techniques, to capture more nuanced attack behaviours. Developing hybrid architectures that combine DCNN's spatial processing with CNN-BiLSTM's sequential modelling could further enhance multi-modal threat detection. Furthermore, investigating the robustness of time-based models against adversarial attacks will be crucial in ensuring their reliability against sophisticated evasion techniques. Extending these models to detect zero-day threats through unsupervised learning on temporal feature clusters could also provide a proactive approach to network security. Addressing these areas will contribute to the development of more resilient and adaptive intrusion detection systems, capable of effectively mitigating evolving cyber threats.

References

[1] S. Aftergood, 'Cybersecurity: The cold war online', 2017.

[2] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, and B. D. Payne, 'Evaluating computer intrusion detection systems: A survey of common practices', *ACM Computing Surveys (CSUR)*, vol. 48, no. 1, pp. 1–41, 2015.

[3] K. B. Virupakshar, M. Asundi, K. Channal, P. Shettar, S. Patil, and D. G. Narayan, 'Distributed Denial of Service (DDoS) Attacks Detection System for OpenStack-based Private Cloud', in *Procedia Computer Science*, 2020. doi: 10.1016/j.procs.2020.03.282.

[4] E. Viegas, A. O. Santin, A. Franca, R. Jasinski, V. A. Pedroni, and L. S. Oliveira, 'Towards an energy-efficient anomaly-based intrusion detection engine for embedded systems', *IEEE Transactions on Computers*, vol. 66, no. 1, pp. 163–177, 2016.

[5] S. Kaur, A. K. Sandhu, and A. Bhandari, 'Investigation of application layer DDoS attacks in legacy and software-defined networks: A comprehensive review', *Int. J. Inf. Secur.*, vol. 22, no. 6, pp. 1949–1988, Aug. 2023, doi: 10.1007/s10207-023-00728-5.

[6] A. H. Lashkari, G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani, 'Characterization of tor traffic using time-based features', in *ICISSP 2017 - Proceedings of the 3rd International Conference on Information Systems Security and Privacy*, 2017. doi: 10.5220/0006105602530262.

[7] Z. Zhao *et al.*, 'DDoS family: A novel perspective for massive types of DDoS attacks', *Computers & Security*, vol. 138, p. 103663, 2024.

[8] J. Mirkovic and P. Reiher, 'A taxonomy of DDoS attack and DDoS defense mechanisms', *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 2, pp. 39–53, Apr. 2004, doi: 10.1145/997150.997156.

[9] M. Zeeshan *et al.*, 'Protocol-based deep intrusion detection for dos and ddos attacks using unsw-nb15 and bot-iot data-sets', *IEEE Access*, vol. 10, pp. 2269–2283, 2021.

[10] X. Fu and E. Modiano, 'Fundamental limits of volume-based network dos attacks', *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 3, no. 3, pp. 1–36, 2019.

[11] A. Thakkar and R. Lohiya, 'A review on machine learning and deep learning perspectives of IDS for IoT: recent updates, security issues, and challenges', *Archives of Computational Methods in Engineering*, vol. 28, no. 4, pp. 3211–3243, 2021.

[12] K. He, D. D. Kim, and M. R. Asghar, 'Adversarial machine learning for network intrusion detection systems: A comprehensive survey', *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 538–566, 2023.

[13] D. Hutchison *et al.*, 'Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition', in *Artificial Neural Networks – ICANN 2010*, vol. 6354, K. Diamantaras, W. Duch, and L. S. Iliadis, Eds, in Lecture Notes in Computer Science, vol. 6354., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 92–101. doi: 10.1007/978-3-642-15825-4_10.

[14] T. M. Mitchell and T. M. Mitchell, *Machine learning*, vol. 1, no. 9. McGraw-hill New York, 1997.

[15] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, vol. 4, no. 4. Springer, 2006.

[16] H. Kheddar, D. W. Dawoud, A. I. Awad, Y. Himeur, and M. K. Khan, 'Reinforcement-Learning-Based Intrusion Detection in Communication Networks: A Review', *IEEE Communications Surveys & Tutorials*, 2024.

[17] V. Hnamte and J. Hussain, 'DCNNBiLSTM: An efficient hybrid deep learning-based intrusion detection system',*Telematics and Informatics Reports*, vol. 10, p. 100053, 2023.

[18] P. Kim, 'Convolutional Neural Network', in *MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence*, P. Kim, Ed., Berkeley, CA: Apress, 2017, pp. 121–147. doi: 10.1007/978-1-4842-2845-6_6.

[19] S. N. Mighan and M. Kahani, 'A novel scalable intrusion detection system based on deep learning', *Int. J. Inf. Secur.*, vol. 20, no. 3, pp. 387–403, June 2021, doi: 10.1007/s10207-020-00508-5.

[20] Y. Fu, Y. Du, Z. Cao, Q. Li, and W. Xiang, 'A Deep Learning Model for Network Intrusion Detection with Imbalanced Data', *Electronics*, vol. 11, no. 6, p. 898, Mar. 2022, doi: 10.3390/electronics11060898.

[21] S. Dasari and R. Kaluri, 'An effective classification of DDoS attacks in a distributed network by adopting hierarchical machine learning and hyperparameters optimization techniques', *IEEE Access*, 2024.

[22] H. Liu and B. Lang, 'Machine learning and deep learning methods for intrusion detection systems: A survey',*applied sciences*, vol. 9, no. 20, p. 4396, 2019.

[23] M. S. Elsayed, N. A. Le-Khac, S. Dev, and A. D. Jurcut, 'DDoSNet: A Deep-Learning Model for Detecting Network Attacks', in *Proceedings - 21st IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2020*, 2020. doi: 10.1109/WoWMoM49955.2020.00072.

[24] J. P. A. Maranhão, J. P. C. da Costa, E. Javidi, C. A. B. de Andrade, and R. T. de Sousa Jr, 'Tensor based framework for Distributed Denial of Service attack detection'*Journal of Network and Computer Applications*, vol. 174, p. 102894, 2021.

[25] T. Khempetch and P. Wuttidittachotti, 'Ddos attack detection using deep learning', *IAES International Journal of Artificial Intelligence*, vol. 10, no. 2, 2021, doi: 10.11591/ijai.v10.i2.pp382-388.

[26] S. Hizal, U. Cavusoglu, and D. Akgun, 'A new Deep Learning Based Intrusion Detection System for Cloud Security', in *HORA 2021 - 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications, Proceedings*, 2021. doi: 10.1109/HORA52670.2021.9461285.

[27] M. A. Ferrag, L. Shu, H. Djallel, and K. K. R. Choo, 'Deep learning-based intrusion detection for distributed denial of service attack in agriculture 4.0', *Electronics (Switzerland)*, vol. 10, no. 11, 2021, doi: 10.3390/electronics10111257.

[28] A. E. Cil, K. Yildiz, and A. Buldu, 'Detection of DDoS attacks with feed forward based deep neural network model', *Expert Systems with Applications*, vol. 169, p. 114520, May 2021, doi: 10.1016/j.eswa.2020.114520.

[29] M. A. Salahuddin, V. Pourahmadi, H. A. Alameddine, M. F. Bari, and R. Boutaba, 'Chronos: DDoS Attack Detection Using Time-Based Autoencoder', *IEEE Transactions on Network and Service Management*, vol. 19, no. 1, 2022, doi: 10.1109/TNSM.2021.3088326.

[30] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, 'Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection', 2018. doi: 10.14722/ndss.2018.23204.

[31] J. Halladay *et al.*, 'Detection and Characterization of DDoS Attacks Using Time-Based Features', *IEEE Access*, vol. 10, pp. 49794–49807, 2022, doi: 10.1109/ACCESS.2022.3173319.

[32] H. Mennour and S. Mostefai, 'Deep learning-based distributed denial-of-service detection',*International Journal of Networking and Virtual Organisations*, vol. 26, no. 1–2, pp. 80–103, 2022.

[33] D. Kumar, R. Pateriya, R. K. Gupta, V. Dehalwar, and A. Sharma, 'DDoS detection using deep learning',*Procedia Computer Science*, vol. 218, pp. 2420–2429, 2023.

[34] H. Aydın, Z. Orman, and M. A. Aydın, 'A long short-term memory (LSTM)-based distributed denial of service (DDoS) detection and defense system design in public cloud network environment', *Computers & Security*, vol. 118, p. 102725, July 2022, doi: 10.1016/j.cose.2022.102725.

[35] Y.-C. Wang, Y.-C. Houng, H.-X. Chen, and S.-M. Tseng, 'Network anomaly intrusion detection based on deep learning approach', *Sensors*, vol. 23, no. 4, p. 2171, 2023.

[36] F. M. Aswad, A. M. S. Ahmed, N. A. M. Alhammadi, B. A. Khalaf, and S. A. Mostafa, 'Deep learning in distributed denial-of-service attacks detection method for Internet of Things networks', *Journal of Intelligent Systems*, vol. 32, no. 1, Jan. 2023, doi: 10.1515/jisys-2022-0155.

[37] T. H. Aldhyani and H. Alkahtani, 'Cyber security for detecting distributed denial of service attacks in agriculture 4.0: Deep learning model', *Mathematics*, vol. 11, no. 1, p. 233, 2023.

[38] A. Ahmim, F. Maazouzi, M. Ahmim, S. Namane, and I. B. Dhaou, 'Distributed denial of service attack detection for the Internet of Things using hybrid deep learning model', *IEEE Access*, vol. 11, pp. 119862–119875, 2023.

[39] R. F. Fouladi, O. Ermiş, and E. Anarim, 'A Novel Approach for distributed denial of service defense using continuous wavelet transform and convolutional neural network for software-Defined network', *Computers and Security*, vol. 112, 2022, doi: 10.1016/j.cose.2021.102524.

[40] F. Hussain, S. G. Abbas, M. Husnain, U. U. Fayyaz, F. Shahzad, and G. A. Shah, 'IoT DoS and DDoS attack detection using ResNet', presented at the 2020 IEEE 23rd International Multitopic Conference (INMIC), IEEE, 2020, pp. 1–6.

[41] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, 'Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy', in *Proceedings - International Carnahan Conference on Security Technology*, 2019. doi: 10.1109/CCST.2019.8888419.

[42] R. Al-Saadi, G. Armitage, J. But, and P. Branch, 'A Survey of Delay-Based and Hybrid TCP Congestion Control Algorithms', *IEEE Commun. Surv. Tutorials*, vol. 21, no. 4, pp. 3609–3638, 2019, doi: 10.1109/COMST.2019.2904994.

[43] G. Ajay and P. Krishnan, 'A Study and Analysis of Effective Data transmission Using UDP',*A-Study-and-Analysis-of-Effective-Data-transmission-Using-UDP. pdf*, 2018.

[44] A. Holzinger, 'Big data calls for machine learning', 2019.

[45] M. Kim, 'Supervised learning-based DDoS attacks detection: Tuning hyperparameters', *ETRI Journal*, vol. 41, no. 5, pp. 560–573, 2019.

[46] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, 'Characterization of encrypted and vpn traffic using time-related', presented at the Proceedings of the 2nd international conference on information systems security and privacy (ICISSP), 2016, pp. 407–414.

[47] D. M. Powers, 'Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation', *arXiv preprint arXiv:2010.16061*, 2020.

[48] G. Somani, M. S. Gaur, D. Sanghi, M. Conti, and R. Buyya, 'DDoS attacks in cloud computing: Issues, taxonomy, and future directions', *Computer communications*, vol. 107, pp. 30–48, 2017.

[49] W. Haider, J. Hu, J. Slay, B. P. Turnbull, and Y. Xie, 'Generating realistic intrusion detection system dataset based on fuzzy qualitative modeling', *Journal of Network and Computer Applications*, vol. 87, pp. 185–192, 2017.

Received: 21/07/2025; Revised: 02/08/2025; Accepted: 04/08/2025; Published: 13/09/2025