ZANCO Journal of Pure and Applied Sciences

TYPE Original article
DOI http://dx.doi.org/10.21271/ZJPAS.37.6.13

OPEN ACCESS

*Corresponding author

Soran Ahmed Hasan
soran.hasan@su.edu.krd

RECEIVED :13 /04 /2025
ACCEPTED :04/08/ 2025
PUBLISHED :31/ 12/ 2025

KEYWORDS:

Anomaly Detection, CICIoT2023 Dataset, DDoS Detection, CNN-LSTM, IoT Security

Enhancing IoT Anomaly Detection using Hybrid CNN-LSTM Model and Interpretable Feature Selection

Soran Ahmed Hasan[1] , Marwan Aziz Mohammed[1,2]

1Department of Software and Informatics Engineering, College of Engineering, Salahaddin University-Erbil, Erbil, Kurdistan Region, Iraq
2Department of Computer Engineering, College of Engineering, Knowledge University, Erbil 44001, Iraq

ABSTRACT

Securing Internet of Things (IoT) networks is an ongoing challenge. As more devices connect to the internet with limited resources, these systems have become more vulnerable to cyberattacks. Many attacks continually evolve and become more sophisticated. This highlights the need for scalable, efficient anomaly detection deployable close to IoT devices to minimize latency, while maintaining high accuracy with low memory and computational demands. Many solutions have been applied for enhancing the problem area, either they are heavy models unsuitable for edge devices or they lack generalizability with recent datasets and current attack traffic patterns. Our research suggests a lightweight anomaly detection model that combines Convolution Neural Network (CNN) and Long Short Term Memory (LSTM) model, to recognize patterns across both spatial and temporal dimensions, as well as identify significant relationships among an interpretable selected set of features. with SHapley Additive exPlanations (SHAP) for feature selection and Synthetic Minority Oversampling Technique - Edited Nearest Neighbors (SMOTE-ENN) for balancing the distribution of classes in the datasets. The model's performance was evaluated using accuracy, precision, recall, and F1 parameters. Following the study, an accuracy rate of 99.12% for multiclassification is achieved in the CICIoT2023 dataset. In the TON_IoT dataset, a multiclassification success rate of 99.08% is reached. The model with 10 features selected achieved 99.0%, 98.85% in the CICIoT2023 and TON_IoT dataset. With just 43,406 trainable parameters and Top 10 features selected proposed framework offers a lightweight, explainable model that is effective for edge IoT devices with limited resources.

1.Introduction

As Denial of Service (DoS), Distributed Denial-of-Service (DDoS), and Mirai Botnet attacks have become more common in recent years, they can cause major challenges if they succeed in breaking down services. This can lead to delays in operations, financial losses, and unauthorized use of sensitive systems. The IoT paradigm has greatly revolutionized numerous sectors through improved communication, real time data exchange, and automation of processes (Shareef, 2023). Applications vary from health monitoring and transportation to home automation and industrial automation, with connected devices enabling real-time decision-making and operation optimization (Neto et al., 2023). In spite of these developments, IoT networks are more vulnerable to cybersecurity attacks owing to their high and heterogeneous connectivity, lightweight protocols, and poor intrinsic security capabilities (Tabassoum et al., 2024) Of these attacks, Distributed Denial of Service (DDoS), through its effect on network availability and integrity by overloading network resources, presents considerable challenges (Hajjouz and Avksentieva, 2024, Hizal et al., 2024). Prior studies underscore the seriousness of IoT vulnerabilities mainly through highlighting strong security frameworks and intrusion detection systems (IDS). The CICIoT2023 dataset fills this gap by providing an encompassing and realistic dataset optimized primarily for IoT security assessment and covering 33 cyberattack types and major classes like DDoS, DoS, Mirai, Reconnaissance, Web-based, Brute Force, and Spoofing (Neto et al., 2023). Numerous approaches have applied diverse methods of machine learning and deep learning models to optimize anomaly detection in IoT environments. Traditional rule-based IDSs are usually ineffective in detecting sophisticated attacks on IoT systems because of their dynamic and heterogeneous nature. To address the limitations of traditional IDS, Machine Learning (ML) algorithms are typically employed these days for IDSs in IoT systems. Nevertheless, these methods are still challenged with the physical and functional diversity of IoT systems(Sanju, 2023).

While ML and DL are being used more frequently in IDS, there is still an obvious gap in developing solutions that are both lightweight enough to be used at the edge and able to keep high accuracy on up-to-date various IoT datasets. Traditional ML approaches using decision trees, random forests, and XGBoost have provided foundational insights (Anwer et al., 2024), but often lack comprehensive effectiveness in capturing temporal and spatial data dependencies, leading to limitations in detection accuracy, particularly against sophisticated attack vectors(Hajjouz and Avksentieva, 2024).Recent research underscores the value of integrating deep learning techniques, such as CNN and LSTM (Hassen and Abdlrazaq, 2024), to capture complex patterns and sequential relationships inherent in network traffic data IoT(Wang et al., 2023a). However, standalone implementations of these models have faced challenges related to computational complexity, redundant feature handling, and limited interpretability (Krzysztoń et al., 2024). The adoption of feature selection methods has gained prominence in recent studies, particularly to address issues of redundant and irrelevant features, which negatively impact model efficiency and accuracy. Techniques such as mutual information, wrapper methods, and hierarchical clustering have been explored to optimize the feature set, significantly improving classifier performance and reducing computational demands (Tabassoum et al., 2024). Nevertheless, few studies have extensively utilized explainable AI-based methods, such as SHAP, specifically for effective feature selection in IoT network security contexts (Hajjouz and Avksentieva, 2024). Addressing these gaps, our research proposes a hybrid CNN-LSTM model integrated with SHAP-based feature selection and targeted data balancing using SMOTE-ENN. Our method is novel as it integrates interpretability, an efficient design, and robust performance across multiple IoT datasets. Addressed the class imbalance issue in recent IoT dataset without deleting any traffic records, thereby preserving rare attacks instances and ensuring a comprehensive detection framework. The SHAP methodology uniquely enhances interpretability by explicitly identifying impactful features, thus improving detection accuracy and model efficiency. The adoption of SMOTE-ENN addresses the inherent class imbalance issues prevalent in IoT datasets, significantly boosting performance on minority classes. Furthermore, our comprehensive evaluation leverages the realistic CICIoT2023 dataset, utilizing a novel class mapping into 14 distinct categories to streamline and enhance the classification process. Furthermore, the model's robustness and generalizability tested on additional ToN-IoT dataset. Therefore, the proposed research provides several distinct contributions to IoT cybersecurity research:

- Utilizing a hybrid CNN-LSTM model optimized for high performance on resource-constrained IoT devices.
- Reducing model complexity by selecting only the top 10 impactful features using (SHAP).
- Addressing class imbalance effectively through SMOTE-ENN balancing to improve detection accuracy across all classes.
- Achieving 99% classification accuracy with fewer computational resources and parameters, making it highly suitable for real-world IoT deployments4.

The structure of this research represented as follows: Section 2 related work, Section 3 describes the Methodology used, including dataset details, data preprocessing techniques, feature selection, and the CNN-LSTM model architecture. Section 4 presents the experimental setup details, including software and hardware specifications. Section 5 discusses the results and provides a detailed performance analysis. Finally, Section 6 concludes the study and suggests directions for future research.

2.Related Work

Anomaly detection in IoT networks has been extensively explored through various machine learning and deep learning techniques, focusing on feature selection, classification models, and balancing techniques to enhance detection accuracy.

1. Binary and Multiclass Classification

Several studies have leveraged deep learning architectures, traditional machine learning approaches, and hybrid models to improve classification performance and address data imbalance challenges.

In (Gueriani et al., 2024), a hybrid CNN-LSTM model was applied for binary classification, the CICIDS2017 dataset was used to confirm that it was generally applicable. The proposed model showed good detection performance across datasets, achieving 98.42% accuracy on CICIoT2023 with 9.17% FPR and 97.46% accuracy on CICIDS2017. The study proposed expanding the model to multiclass classification, adding attention-based Transformer based attention mechanisms, and deploying on real-time hardware (Raspberry Pi & FPGA).

Similarly (Neto et al., 2023) explored Logistic Regression, Perceptron, Adaptive Boosting, Random Forest, and Deep Neural Networks (DNN) for binary, 8-class, and 34-class classification used CICIoT2023 dataset, reporting 99.68% accuracy (RF, binary), with significantly lower scores for multi-class cases. The study suggested optimizing machine learning models and investigating feature influence for improved transferability across datasets. demonstrates how the CICIoT2023, with its wide topology and variety of attack scenarios, offers a realistic and thorough benchmark for creating and assessing ML-based intrusion detection and classification techniques.

(Nazir et al., 2024) used Principal Component Analysis (PCA) with a Hybrid CNN-LSTM model for binary classification, achieving 99.9% accuracy across multiple datasets, with future recommendations on optimizing the model for edge deployment using quantization and pruning techniques. Several studies have explored hybrid deep learning models for multi-class classification.

(Khan and Alkhathami, 2024) analyzed binary, 8-class, and 34-class classification using Random Forest (RF), Adaptive Boosting (AB), Logistic Regression (LR), Perceptron (PER), and Deep Neural Networks (DNN), achieving 99.55% accuracy for binary classification, and slightly lower for multi-class (95.54% for 8-class, 96.32% for 34-class). The study proposed expanding attack class analysis (e.g., Spoofing & Recon attacks) and integrating security solutions into real-world IoT healthcare applications.

Traditional machine learning methods with feature selection have also demonstrated high performance (Hajjouz and Avksentieva, 2024) employed Spearman correlation and hierarchical clustering with CatBoost for multi-class classification (19 classes) on CICIoT2023 dataset,

with preprocessing that reducing the feature set into the most important 23 features achieving 99.96% accuracy and proposing real-time deployment as future work. The study establishes a new cybersecurity benchmark by providing detailed protection mechanisms against advanced threats, marking significant progress in network security. This method provides companies with a more efficient tool to identify and minimize cyber risks.

2- Feature Selection methods

Feature selection and deep learning optimization techniques have been widely explored (Khanday et al., 2024) introduced feature elimination and duplication removal , The process involves selecting the top twenty features from the CICIoT2023 dataset using an Extra Tree Classifier. with DNN, CNN, and LSTM for binary, 3-class, and 12-class classification, reporting 94.96% accuracy for binary classification and up to 92.73% for 12-class scenarios. The study emphasized optimizing deep learning models for IoT security and real-time deployment.

(Tabassoum et al., 2024) suggest Mutual Information + GAN-based feature selection, it reduces 47 features to 20 before using an RNN model for multi-class classification (16 attack classes: DDoS & DoS), improving accuracy from 96% to 97%, and proposed comparing its feature selection method with traditional techniques while testing scalability on larger datasets.

Further research has incorporated feature selection techniques to enhance model efficiency.(Ji et al., 2024) introduced SelectKBest + Mutual Information (SelectKBest-MI) for feature selection, applying CNN-SVM-GWO using CIC-IoT-2023 dataset for binary classification, achieving 99.60% accuracy with Random Forest (RF) and 99.49% with XGBoost (XGB), precision 0.99, recall 0.99, F1- score 0.99 with recommendations to extend their approach to multi-class classification.

3. Comparative Perspective and Our Contribution

Compared to these studies, our proposed framework introduces a SHAP based feature selection method, which improves explainability over traditional feature selection techniques like Mutual Information, Pearson Correlation, and Spearman Correlation used in (Hajjouz and Avksentieva, 2024, Ji et al., 2024, Modi) While most studies rely on Random Forest, XGBoost, or CatBoost, our hybrid CNN-LSTM model effectively captures both spatial and temporal dependencies in attack traffic, providing a more robust classification strategy. Furthermore, our SMOTE-ENN approach selectively oversamples five minority classes, addressing imbalance issues more efficiently than class weighting (Hajjouz and Avksentieva, 2024) or standard oversampling (Khanday et al., 2024). As shown in **Table 1**. Our model achieves accuracy of 99.12% with a high weighted average accuracy of 99.06% and a macro average accuracy of 98.26%, demonstrating superior generalization across multiple attack types and making it a more scalable and interpretable solution for IoT security.

Table 1. Summary of Related Works – Strengths and Limitations

Study	Model / Technique	Dataset(s)	Key Strengths	Accuracy	Limitations
(Gueriani et al., 2024)	CNN-LSTM	CICIoT2023, CICIDS2017	Good cross-dataset evaluation (CICIoT2023 & CICIDS2017), hybrid DL model	98.42% (CICIoT2023) 97.46% (CICIDS2017)	Focused on binary classification only, high FPR (9.17%)
(Neto et al., 2023)	ML classifiers RF, AB, LR, PER, DNN	CICIoT2023	Test multiple ML classifiers in Real time recent Dataset, with high binary result	RF 99.68% (binary), ~99% (multi-class)	Imbalance in class distribution, feature selection not used, not memory efficient used Tree based model, Edge feasibility is not

					estimated.,
(Khanday et al., 2024)	Extra Trees + LSTM, CNN	CICIoT2023	Uses 20 optimal features from CICIoT2023 Trains on simplified feature set for lightweight deployment	1D-CNN achieved an accuracy of 99.87%(binary) LSTM 92% (binary)	without AI explainability method, not generalize to other IoT datasets, heavy computational model
(Tabassoum et al., 2024)	RNN + GAN + MI	CICIoT2023	20 feature selected, Improved accuracy using GAN-MI based FS, effective against adversarial data	97% (real data) 93% (synthetic data)	Limited to DDoS/DoS attacks, not generalize to other IoT datasets, Complexity of GANs
(Nazir et al., 2024)	CNN-LSTM + PCA	IoT-23, N-BaIoT, CICIDS2017	High accuracy (99.9%), discussed edge deployment (future), optimization techniques (quantization, pruning)	IoT-23: 95% N-BaIoT: 99.99% CICIDS2017: 98.99%	PCA reduces model interpretability, Not feature selection mechanism, Not used recent dataset, High computational complexity, Edge feasibility is not estimated
(Khan and Alkhathami, 2024)	ML + DNN	CICIoT2023	Addressed multiple class subset (binary–34 class)m SMOTE algorithm for balancing, reduced feature space to 31 by Pearson correlation coefficient.	(RF) 99.55% (binary), ~96% (multi-class)	Lower performance on multiclass tasks (95.5–96.3%), Limited interpretability, High computational complexity, generalization issues
(Hajjouz and Avksentieva, 2024)	Hierarchical Feature Selection + CatBoost	CICIoT2023	High accuracy (99.96%), good FS strategy	99.96% (multi-classes)	computational cost of preprocessing and tuning, generalization issues, edge feasibility issue, discarding traffic of rare attacks, not capture complex pattern of evolving attacks
(Ji et al., 2024)	CNN-SVM-GWO, RF, XGB	CICIoT2023	Very high binary accuracy, robust evaluation	99.60% (RF), 99.49% (XGB)	Binary classification only, recommend extending to multiclass, Not memory efficient used Tree based model, Edge feasibility is not estimated
Proposed	**CNN-LSTM + SHAP + SMOTE-ENN**	**CICIoT2023, ToN-IoT**	**Multiclass generalization (CICIoT2023 & ToN-IoT), SHAP for explainability, edge deployment feasibility with 43K params** **10 Top Features selected,** **New balancing strategy to address ciciot2023 imbalance issue**	**99.12% CICIoT2023 (14 classes)** **99.08% ToN-IoT(10 classes)**	**Currently focused on supervised learning. Adversarial robustness, adaptive learning , model quantization and pruning techniques not addressed (future work)**

3.Methodology

The (IoT) and its applications become increasingly used by people, as they enhance convenience in daily life. Due to its popularity, attacks targeting these devices have surged significantly, potentially rendering the entire system unavailable. The attacks include Distributed Denial-of-Service (DDoS), Denial-of-

Service (DoS), Mirai variations, Brute Force, Spoofing, Reconnaissance, Man-in-the-Middle, and Web-based attacks. Consequently, as the frequency of attacks has increased, the techniques for detecting malware in the IoT have likewise increased.

To effectively detect various types of attacks, the anomaly detection model needs to be trained on up-to-date, balanced datasets that feature many attack types and a wide range of realistic traffic patterns for each type. Therefore, proposed research utilized a publicly available, real-world IoT intrusion detection dataset.

This section outlines the complete workflow of the proposed anomaly detection framework. As illustrated in **Figure 1** (the framework flowchart) and detailed in Pseudocode as shown in **Algorithm 1**, the framework is composed of several key stages: data preprocessing, class balancing, feature selection, model training, and performance evaluation. The overall architecture leverages the CICIoT2023 dataset and integrates explainable AI and deep learning components for multiclass classification of IoT network intrusions.

Algorithm 1. CNN-LSTM Multiclass Anomaly Detection Using SHAP and SMOTE-ENN

```
Input: CICIoT2023 Dataset & TON-IoT Dataset with N features
Output: Classification performance metrics (Accuracy, Precision, Recall, F1-score)

Define:

    • data_cleaned: preprocessed dataset
    • data_balanced: SMOTE-ENN balanced dataset
    • selected_features: top 10 important features from SHAP
    • reshaped_data: input reshaped for CNN-LSTM model

procedure CNN_LSTM_Multiclass()

  Step 1: Preprocessing
    data_cleaned ← preprocess the raw dataset (handle missing values, encode categories, standardize)

  Step 2: Handle Class Imbalance
    data_balanced ← apply SMOTE-ENN to data_cleaned

  Step 3: Split Dataset
    (train, validation, test) ← split data_balanced into 70%, 15%, and 15% respectively

  Step 4: Feature Selection Using SHAP
    baseline_model ← train CNN-LSTM using all features
    shap_scores ← compute SHAP values on baseline model
    selected_features ← select top 10 features based on SHAP

  Step 5: Prepare Final Dataset
    train_selected ← extract and reshape train using selected_features
    test_selected ← extract and reshape test using selected_features

  Step 6: CNN-LSTM Model Training
    model ← define CNN-LSTM architecture
    compile model with Adam optimizer and categorical crossentropy loss
    train model using train_selected (epochs = 20, batch size = 128)

  Step 7: Evaluation
    predictions ← model.predict(test_selected)
    metrics ← evaluate predictions using accuracy, precision, recall, and F1-score

end procedure
```

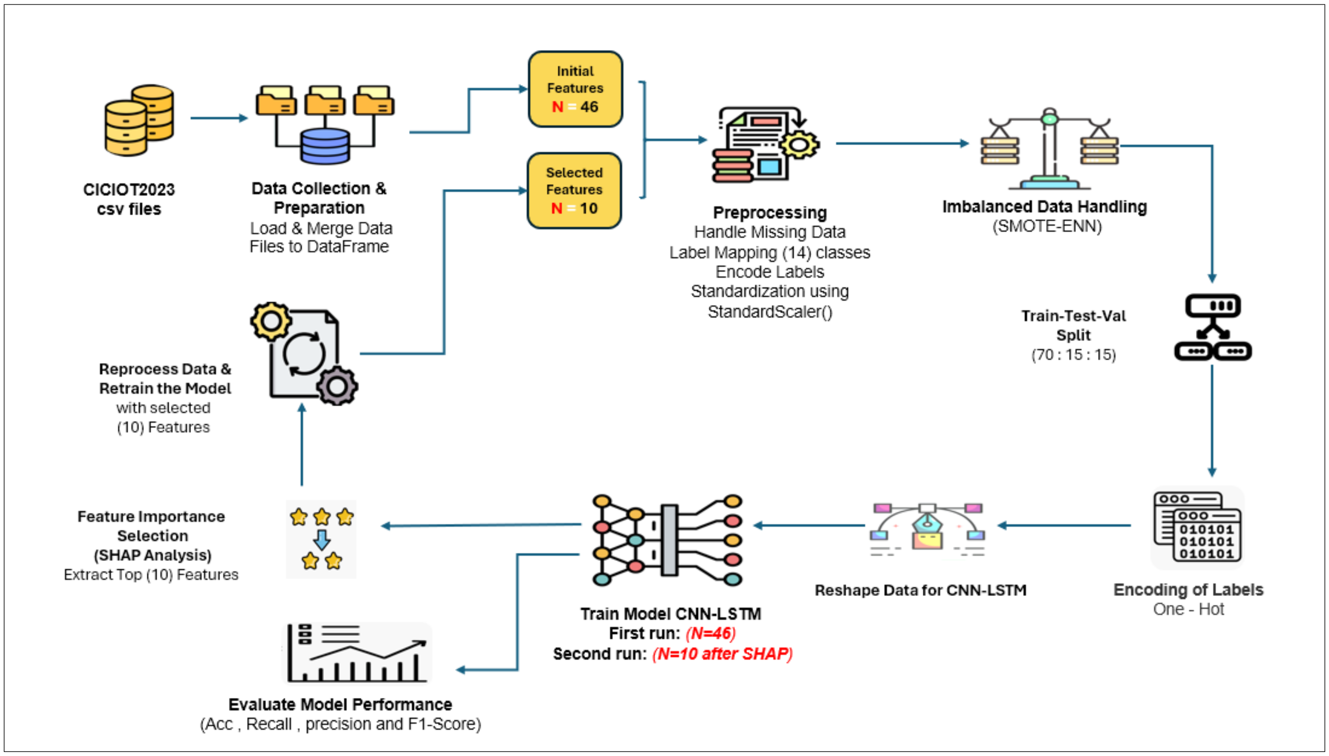


Figure 1. Flow Diagram of Proposed work

3.1 Dataset Description

In this section, information about CICIoT2023 and TON_IoT datasets used in the study is provided. Afterwards, the preprocessing steps of the dataset are explained. Then, the deep learning algorithms used in the study are defined. CICIoT2023 dataset was produced by (Neto et al., 2023), a real time intrusion detection dataset designed for IoT security evaluation, Developed by the Canadian Institute for Cybersecurity (CIC), it provides a comprehensive benchmark for detecting cyber threats in IoT environments, generated from a large scale IoT network consisting of 105 devices, including smart home systems, cameras, sensors, and microcontrollers. Unlike traditional datasets, CICIoT2023 captures real-world attack scenarios where malicious IoT devices directly target other devices, making it highly relevant for cybersecurity research. It includes 33 distinct attack types, categorized into seven groups DDoS, DoS, Reconnaissance, Web-based, Brute Force, Spoofing, and Mirai-based attacks, as detailed in **Table 2**. Network traffic was captured using Wireshark and converted into structured formats such as (CSV and pcap) to facilitate analysis. The dataset contains critical network flow features like flow duration, packet rates, protocol types, and statistical metrics, providing valuable insights for machine learning based intrusion detection systems. With its comprehensive representation of both benign and malicious traffic, the dataset serves as a reliable resource for evaluating feature selection methods and classification models, as demonstrated in recent studies.

The dataset can be accessed via the following URL.

CICIOT2023 URL:

https://www.unb.ca/cic/datasets/iotdataset-2023.html

TON-IOT URL:

https://research.unsw.edu.au/projects/toniot-datasets

Table 2. CICIoT2023 Attacks Distribution (Neto et al., 2023)

Class	Attack	Rows
DDoS	Ack Fragmentation	285,104
	Udp Flood	5,412,287
	Slowl0ris	23,426
	Icmp Flood	7,200,504
	Rstfin Flood	4,045,285
	Pshack Flood	4,094,755
	Http Flood	28,790
	Udp Fragmentation	286,925
	Icmp Fragmentation	452,489
	Tcp Flood	4,497,667
	Syn Flood	4,059,190
	Synonymouslp Flood	3,598,138
Dos	Tcp Flood	2,671,445
	Http Flood	71,864
	Syn Flood	2,028,834
	Udp Flood	3,318,595
Recon	Ping Sweep	2262
	Os Scan	98,259
	Vulnerability Scan	37,382
	Port Scan	82,284
	Host Discovery	134,378
Web-Based	Sql Injection	5245
	Command Injection	5409
	Backdoor Malware	3218
	Uploading Attack	1252
	Xss	3846
	Browser Hijacking	5859
Brute-Force	Dictionary Brute Force	13,064
Spoofing	Arp Spoofing	307,593
	Dns Spoofing	178,911
Mirai	Greip Flood	751,682
	Greeth Flood	991,866
	Udpplain	890,576
	Total	**45,588,384**

3.2 Data Preprocessing

Using datasets in deep learning algorithms without preprocessing is not appropriate. in deep learning training the string values in the dataset, must be converted to numerical values and clean Preprocessing aims to provide the algorithm with smoother data, thereby enhancing the efficiency of the model as shown in **Figure 2**. The initial CICIoT2023 dataset contained approximately 46 million records, from which a subset of 2,366,956 records was extracted for proposed research, before proceeding with analysis the missing and incomplete values were eliminated to ensure the integrity of the data. This cleaning process was critical to avoid any potential biases or disruptions during model training and evaluation.

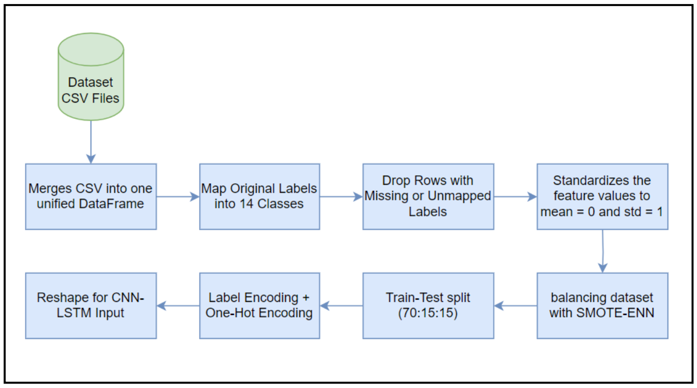

Dataset CSV Files
Merges CSV into one unified DataFrame
Map Original Labels into 14 Classes
Drop Rows with Missing or Unmapped Labels
Standardizes the feature values to mean = 0 and std = 1
balancing dataset with SMOTE-ENN
Train-Test split (70:15:15)
Label Encoding + One-Hot Encoding
Reshape for CNN-LSTM Input

Figure 2. Dataset Preprocessing steps.

3.2.1 Class Label Mapping

To make the classification process more efficient attack categories in the dataset were grouped into 14 specific classes, ensuring a well structured and balanced representation of data. The remapped class distribution shown in **Table 3**, organizes attacks into meaningful categories to improve classification performance. This categorization simplifies detection by clustering similar attack types, Specifically, the Not-DDoS class was introduced to include all attack types not classified under DoS, DDoS, or Mirai, Not-DDoS class encompasses Reconnaissance (Recon), Web-Based attacks, Brute Force attacks, and Spoofing, which differ from volumetric attacks as they focus on unauthorized access attempts, system exploitation, and identity impersonation, rather than flooding network resources, By consolidating these diverse non-DDoS threats into a single category, the classification model becomes more effective at differentiating between volumetric attacks (DoS, DDoS, Mirai) and non-DDoS attack behaviors, leading to improved detection accuracy and model generalization.

Table 3. Subset of CICIoT2023 Dataset used

Class	Before Smote-Enn	After Smote_Enn
DoS	409,683	402461
DDoS ICMP Flood	364,557	364048
DDoS UDP Flood	274,432	272121
DDoS TCP Flood	228,873	227649
DDoS PSHACK Flood	207,971	207408
DDoS SYN Flood	206,146	204752
DDoS RSTFIN Flood	204,892	204661
DDoS SynonymousIP Flood	182,094	181627
Mirai	133,220	132183
Benign	55,859	132113
Not-DDoS	44,587	131885
DDoS ICMP Fragmentation	22,890	131605
DDoS UDP Fragmentation	14,611	95152
DDoS ACK Fragmentation	14,498	92669
TOTAL	**2,364,313**	**2,780,334**

3.2.2 Encoding Categorical Variables

Converting category values into numerical values is referred to label encoding, a popular method utilized in deep learning. Label encoding involves assigning a numerical value to each unique category to enable efficient algorithm processing. For instance, absolute feature values such as HTTP Flood, UDP Flood, and TCP Flood can be encoded using label encoding method. Label encoding assigns corresponding numerical values to each of these features. Another common approach for transforming categorical data into numerical form is One-Hot encoding, which represents categorical variables as binary vectors (Bakhsh et al., 2023) Since machine learning models require numerical inputs, the categorical attack labels were one-hot encoded. Encoding method ensures that the model does not mistakenly interpret categorical labels as ordinal values. Encoding transformation improves learning stability and prevents biases during training.

3.2.3 Feature Standardization

Every feature in the original dataset has a different data range, The standardization approach used to scale features and confirming that the data followed a normal distribution so that the model's accuracy and speed of convergence improved (Wang et al., 2023a), The standardization changed the original data's(mean) to 0 and its standard (deviation) to 1, The StandardScaler technique was applied to standardize numerical values as defined in equation (1).

$$x' = (x - \mu)/\sigma \qquad (1)$$

where $\mathbf{x}$ is the original value $\boldsymbol{\mu}$ is the mean and $\boldsymbol{\sigma}$ is the standard deviation of the feature values. Standardization ensures that all features contribute equally to the classification process and enhances detection accuracy by preventing features with larger magnitudes from dominating learning patterns, By implementing these preprocessing steps the dataset was prepared for effective intrusion detection, ensuring that both feature selection and model training were performed on clean, well-structured, and normalized data.

3.3 Imbalanced Data Handling

Addressing class imbalance and consistence distribution of classes in dataset is crucial for reliable predictions (Khan and Alkhathami, 2024). To address that SMOTE-ENN was used to oversample minority classes and remove noise of majority samples. SMOTE creates synthetic data for underrepresented classes, while ENN cleans ambiguous instances, resulting in a balanced dataset that improves model accuracy and classification performance. Two underrepresented classes DDoS HTTP Flood and DDoS Slow Loris were excluded, which were insufficient for reliable oversampling without risking model overfitting on huge synthetic data. With balancing the dataset achieved a more uniform class distribution, enabling the model to learn effectively from all attack types and improving its overall robustness in IoT intrusion detection.

3.4 Feature Selection Using SHAP Analysis

To enhance model performance and interpretability, the SHAP method was applied to identify the most influential features contributing to model prediction (Shtayat et al., 2023). Generated a ranking of 20 features based on their contribution to the classification process, as shown in **Figure 3**. The features such as Inter-Arrival Time (IAT), Protocol Type, SYN flag count, and TCP & ICMP attributes had the largest impact on model decision making, Particularly DDoS-TCP_Flood, DoS, and UDP-based attacks. To refine the model and reduce computational complexity, Top 10 most impactful features were selected, feature selection makes eliminate redundant or less significant attributes, ensuring that the final model focuses on the most informative network traffic characteristics. Thereby improves model efficiency without compromising accuracy (Albulayhi et al., 2022). Top 10 features were used in the final model training phase, contributing to a more optimized and interpretable intrusion detection framework.

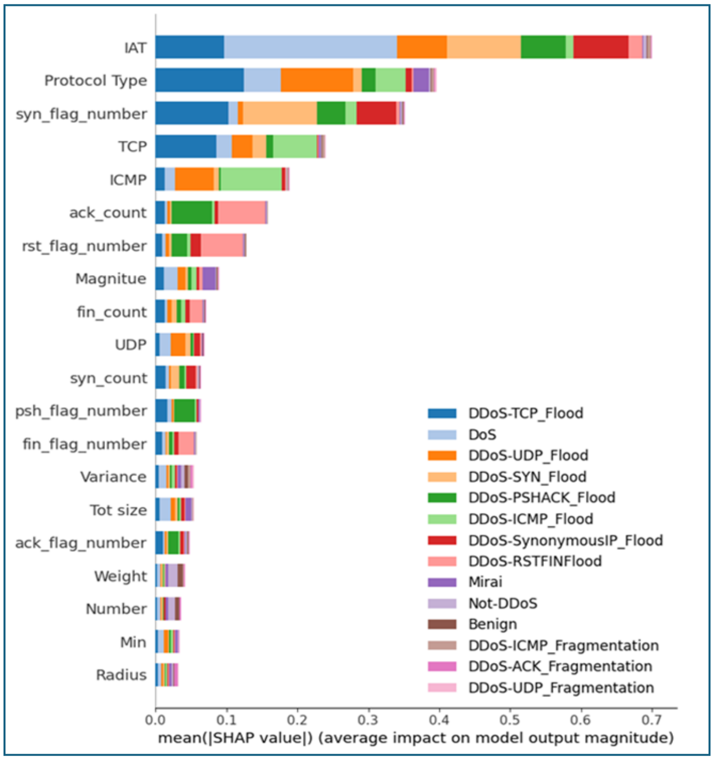


Figure 3. SHAP Analysis for CNN-LSTM Model Predictions using CICIoT2023

3.5 TON_IoT

The IoT Lab at the UNSW Canberra Cyber Institute collected the TON_IoT dataset in 2019 in a real-world, large-scale test environment (Moustafa, 2021). The collection includes modern IoT attacks like scanning, DoS, DDoS, ransomware, backdoors, injection, Cross site scripting (XSS), password cracking, and Man-In-the-Middle (MITM) attacks. The Train-Test Network dataset, a subset of the TON_IoT, is utilized in the research. Network dataset contains a CSV file that includes a subset with (461043) records of the entire attack types and normal records. The selected subset suggested to be used for evaluating new AI-based cybersecurity solutions and making fair comparisons between the new security solutions. The same CICIoT2023 dataset preprocessing steps applied also to selected TON-IoT set, The Train-Test Network dataset is already balanced except MITM class we oversampled it by SMOT-ENN.

3.6 Model Architecture

In this work, a hybrid of CNN-LSTM networks is used to improve detection, The CNN part extracts spatial information and the relationship between the input features, while the LSTM learns long-term dependencies between them. The CNN-LSTM structure was designed via empirical assessment and iterative enhancement. Various model structures, from complex to light, were manually evaluated by adding and removing different 1D CNN and LSTM layers. After the target structure that fit our goal was selected, hyperparameter tuning was applied by altering the number of convolutional filters from (256, 128, 64), kernel sizes (3, 5), LSTM units (64, 96), and dropout rates. The fusion of ML and DL approaches enables better classification of network traffic data (Alzahrani et al., 2024, Manokaran and Vairavel, 2024). The blocks proposed in the CNN-LSTM model are separated according to the structure from bottom to top as shown in **Figure 4**.

- Input Layer: Defines the input shape based on the top 10 selected features, resulting in a shape of (10,1).
- Conv1D Layer: A one-dimensional convolutional layer with 64 filters and a

kernel size of 3, designed to capture local patterns from sequential data.

- MaxPooling1D Layer: Reduces dimensionality by summarizing convolutional outputs, helping to extract key features and mitigate overfitting.
- LSTM Layer: Contains 64 units, responsible for modeling dependencies between feature representations extracted by CNN. The LSTM improves learning by capturing contextual relationships that CNN alone might overlook.
- Flatten Layer: Converts the LSTM output into a one-dimensional vector, preparing it for the fully connected layers.
- Fully Connected Dense Layer: A 128-neuron layer that refines and combines extracted features for better classification.
- Dropout Layer: Applied with a rate of 0.3 to prevent overfitting by randomly deactivating a fraction of neurons during training, ensuring better generalization.
- Output Layer: A dense layer with a SoftMax activation function, classifying inputs into one of 14 distinct attack categories.

To ensure compatibility with both CNN and LSTM layers, the input data was reshaped from a two-dimensional format to a three dimensional array with dimensions (batch size, 10, 1). This transformation facilitates spatial feature extraction by CNN while also allowing the LSTM to model relationships within the extracted features. The model was trained using the parameters as shown in **Table 4.**

Table 4. Model Hyperparameters

Hyperparameter	Value	Description
Learning Rate	0.001	Controls the step size during weight updates
Batch Size	128	Number of samples per weight update
Epochs	20	Number of full passes through the training set
Loss Function	Categorical Crossentropy	Loss function used for multiclass classification
Optimizer	Adam	Adaptive moment estimation optimizer
Activation Function	ReLU / Softmax	ReLU for hidden layers, Softmax for output layer

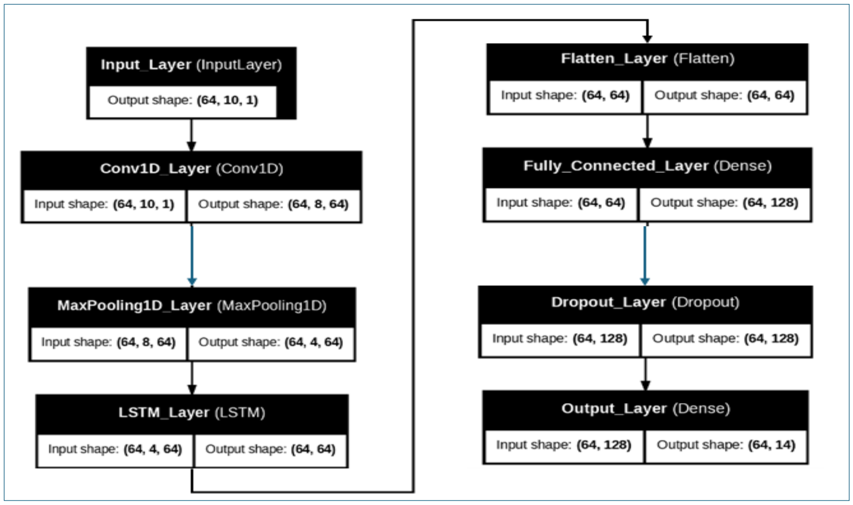


Figure 4. Architecture of proposed hybrid (CNN+LSTM) model.

4. Experimental Setup

The experiments and model training were conducted using the Kaggle Notebook environment, which provides cloud-based computational resources for deep learning tasks. The hardware and software configurations used in this research represented in **Tables [5,6].** The proposed model was trained using GPU resources from the Kaggle Notebook environment. We also trained on a CPU to compare, and as expected, using a GPU reduced training time significantly, which made it easier to experiment and iterate. Allow us to apply several hyperparameter tunings and model structures to achieve an efficient model.

Table 5. Hardware Configuration

Component	Specification
CPU	Intel® Xeon® Processor @ 2.00 GHz
RAM	32 GB (32,873,392 kB)
GPU	Dual NVIDIA Tesla T4 GPUs, each 15 GB VRAM
CUDA Version	12.6
GPU Driver Version	560.35.03

Table 6. Software Environment

Software / Library	Version / Usage
Python	Version 3.10.12 (GCC 11.4.0)
TensorFlow	Version 2.17.1 (Deep Learning Model Training)
Scikit-learn	Data Preprocessing & Evaluation Metrics
SHAP	Explainable AI-based Feature Selection
NumPy & Pandas	Data Handling and Manipulation

5. Results and Analysis

This section presents the experimental results obtained from training and evaluating the proposed CNN-LSTM model using both the complete set of 46 features and the SHAP selected 10 most important features from the CICIoT2023 and TON_IoT dataset. The analysis includes a comparative evaluation of the model performance based on various metrics such as accuracy, precision, recall, F1-score, and visual tools including confusion matrices and accuracy learning curves **Table 7.**

Table 7 Evaluation Metrics

Metric	Equation	Description
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	finds the percentage of cases that are correctly classified through of all instances.
Precision	$\frac{TP}{TP+FP}$	Finds the percentage of correctly predicted positive cases among all predicted positives.
Recall	$\frac{TP}{TP+FN}$	evaluates how well the model can detect actual positive cases.
F1-Score	$2*\frac{\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}}$	Represents the precision and recall harmonic meaning, which is especially helpful when dealing with class imbalance.

These metrics provide a comprehensive assessment of the model's ability to classify network traffic across multiple attack categories. Note: TP (True Positive), TN (True Negative), FP (False Positive), FN (False Negative). Along with accuracy metrics, we evaluated computational trade-offs to show how the small feature set is more effective. The SHAP-selected method, as shown in **Table 8**, that the SHAP-selected 10-feature model reduces GPU training time from 102 seconds to 93 seconds and inference time from 0.000082 seconds to 0.000067 seconds per sample. The drop was even more noticed on the CPU due to hardware architecture and how parallelism ability. These results show that reducing features can make things run more smoothly, especially when they are deployed in real time or on edge devices.

Table 8. Impact of Feature Selection on Training and Inference Time Across Hardware Platforms

	Training using GPU source		**Training using CPU source**	
Model on CICIOT2023 dataset	Time per epoch (seconds)	Inference time per sample (seconds)	Time per epoch (seconds)	Inference time per sample (seconds)
Dataset with 46 Features	102	0.000082	449	0.000182
Dataset with Selected 10 Features	93	0.000067	144	0.000084

5.1 Performance Metrics Comparison

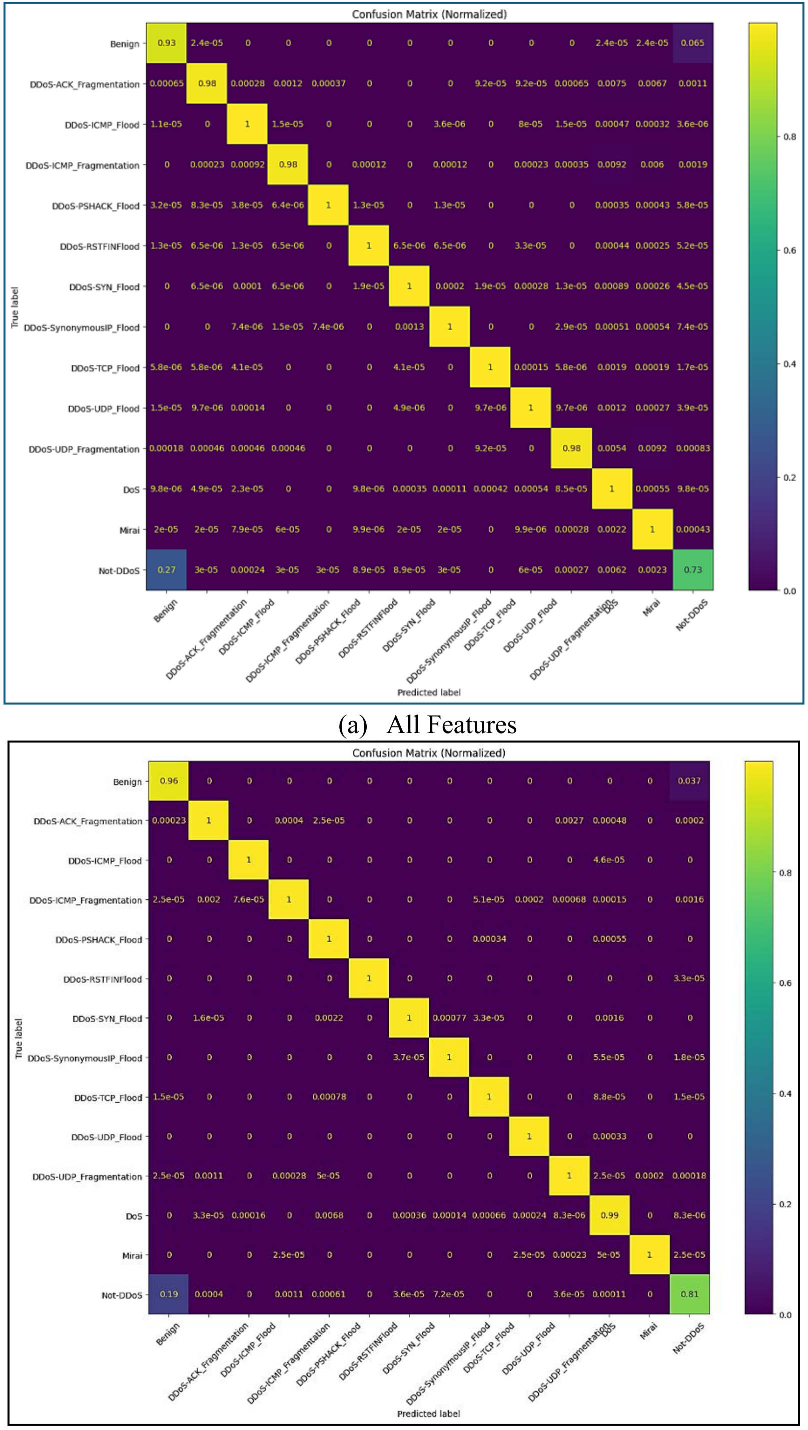
The model was first trained using all features to establish a performance baseline. Then, SHAP-based feature selection was applied to reduce the feature set to the top 10, followed by retraining under identical conditions. **Table 9**. presents the classification metrics (precision, recall, and F1-score) for each of the 14 classes in both scenarios. The proposed model achieved high performance in both configurations, with slightly improved precision and F1-scores after feature selection in many classes. Notably, minority classes such as Not-DDoS and Benign showed marked improvement in F1-score after feature selection. The weighted average F1-score achieved was 0.9911 for all features and 0.9899 for the 10 selected features, indicating only a marginal drop in overall accuracy but enhanced computational efficiency and interpretability.

Table 9. Comparison of the model's precision, recall, and F1-score across all classes using all 46 features versus the reduced 10-feature set in CICIoT2023

Class Name	Precision (46)	Precision (10)	Recall (46)	Recall (10)	F1-Score (46)	F1-Score (10)
Benign	0.8146	0.8347	0.9348	0.9631	0.8706	0.8943
DDoS-ACK_Fragmentation	0.9957	0.9965	0.9813	0.9960	0.9885	0.9962
DDoS-ICMP_Flood	0.9996	0.9998	0.9991	1.0000	0.9993	0.9999
DDoS-ICMP_Fragmentation	0.9980	0.9985	0.9809	0.9951	0.9894	0.9968
DDoS-PSHACK_Flood	1.0000	0.9837	0.9990	0.9991	0.9995	0.9913
DDoS-RSTFINFlood	0.9999	1.0000	0.9992	1.0000	0.9995	1.0000
DDoS-SYN_Flood	0.9980	0.9992	0.9982	0.9954	0.9981	0.9973
DDoS-SynonymousIP_Flood	0.9995	0.9988	0.9975	0.9999	0.9985	0.9993
DDoS-TCP_Flood	0.9992	0.9985	0.9977	0.9991	0.9985	0.9988
DDoS-UDP_Flood	0.9987	0.9995	0.9983	0.9997	0.9985	0.9996
DDoS-UDP_Fragmentation	0.9917	0.9963	0.9829	0.9981	0.9873	0.9972
DoS	0.9943	0.9983	0.9978	0.9916	0.9960	0.9949
Mirai	0.9909	0.9998	0.9969	0.9996	0.9939	0.9997
Not-DDoS	0.8932	0.9523	0.7251	0.8053	0.8005	0.8726
Macro Avg	**0.9767**	**0.9826**	**0.9706**	**0.9816**	**0.9727**	**0.9813**
Weighted Avg	**0.9915**	**0.9906**	**0.9912**	**0.9900**	**0.9911**	**0.9899**

5.2 Confusion Matrix Analysis

We conducted a qualitative analysis of the misclassifications using the normalized confusion matrix as shown in **Figure 5**. Our results demonstrate that most classes, particularly DoS and DDoS variants, were classified with high recall and precision because of their unique massive characteristics, for example fragmented payloads, high packet rates. The most significant misclassifications, however, happen between the broad Not-DDOS class and Benign (~19% misclassified as Not-DDOS). This is to be expected since the Not-DDOS class combines a wide range of attacks including malware, browser hijacks, SQL injections, and reconnaissance scans that overlap with benign traffic patterns. Additionally, some DDoS types with similar statistical characteristics as DDoS-ICMP_Flood vs. DDoS-ICMP_Fragmentation displayed slight confusion (~0.002–0.01).



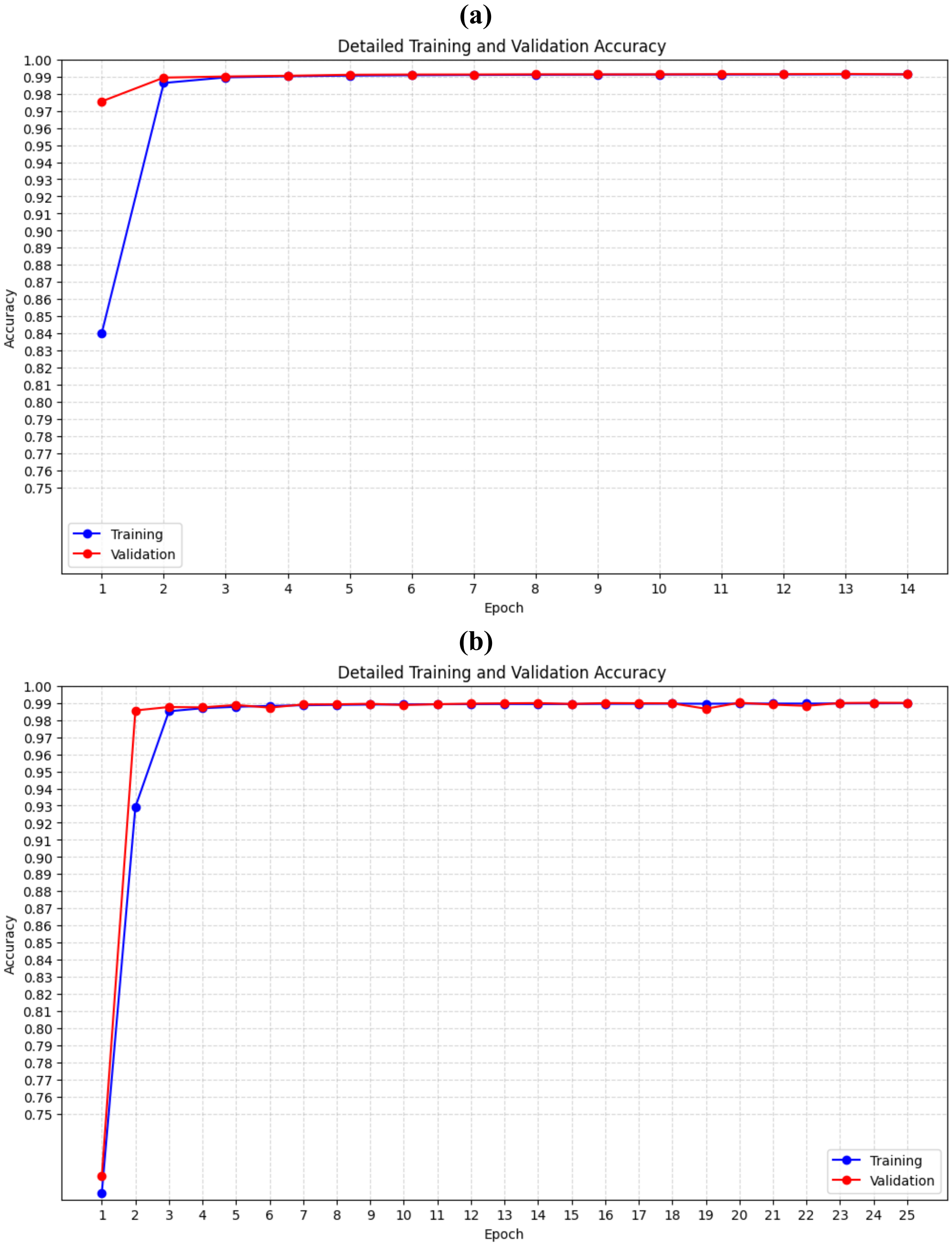
(a) All Features

(b) Select 10 Features

Figure 5(a,b). Normalized confusion matrix shows the classification results across all attack classes and Benign traffic. Diagonal elements represent correct classifications, while off-diagonal elements highlight misclassifications between classes in CICIoT2023

5.3 Training and Validation Accuracy

Training and validation accuracy curves illustrate the dynamics of the model. The convergence behavior across epochs confirms the model's robustness and capacity to generalize. As shown in **Figure [6,7]** both training setups converged rapidly, reaching stability within the first few epochs. The validation accuracy remained consistently high (99%) throughout training, suggesting that the model did not suffer from overfitting and learned effectively from the data.


(a)

(b)

Figure 6 training performance **a.** All features, **b.** Selected 10 features in CICIoT2023

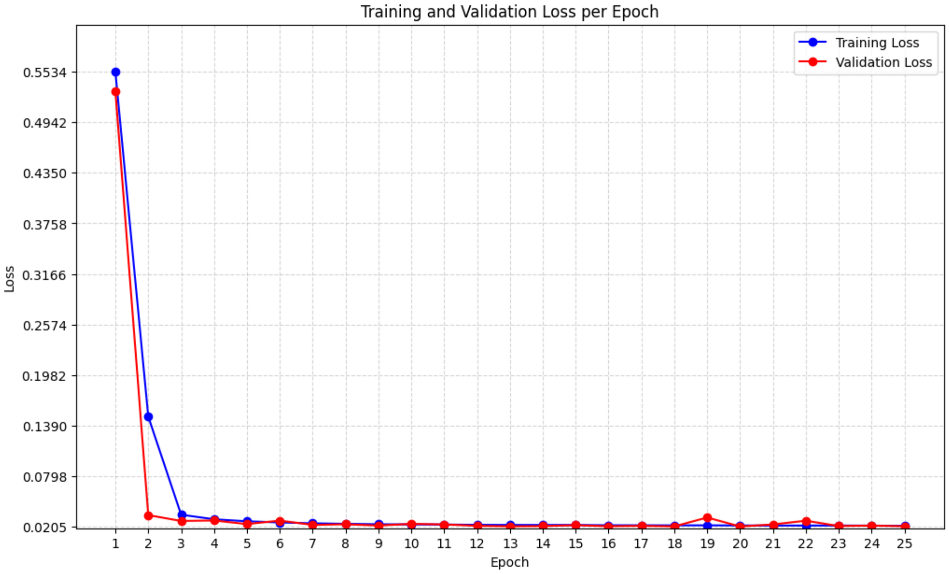
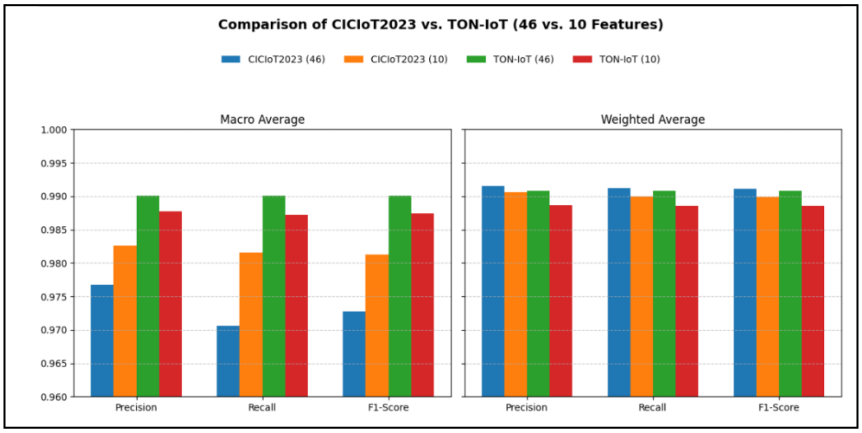


Figure 7. Loss Function of CICIoT2023

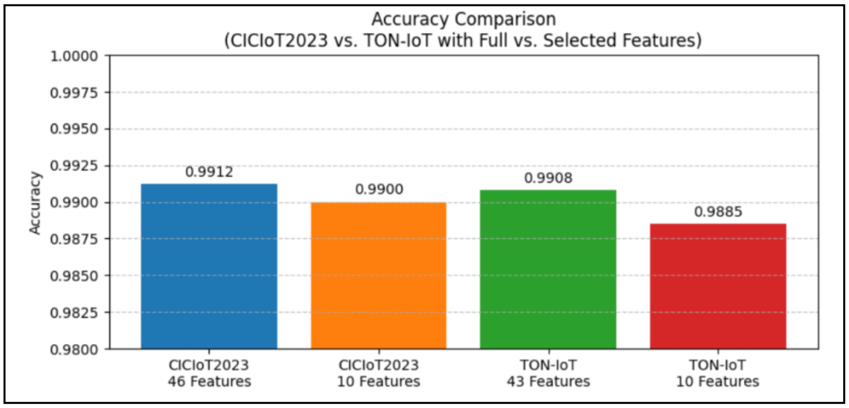
5.4 Model Validation Across Multiple Dataset

To determine the robustness and generalizability of model, a comprehensive test performed using two different datasets, CICIot2023 and TON_IoT, Both the all-feature sets (46 for CICIoT2023 and 43 for TON_IoT) and a reduced selected set of 10 features each were used to evaluate the performance. A comparison based on macro and weighted averages for F1-Score, Precision, and Recall is shown in **Figure 8.a.** The models tested on the TON-IoT dataset consistently perform better than those on the CICIoT2023 dataset in both averaging techniques. Interestingly, the TON-IoT model with its complete feature set performs better in the macro-average recall and F1-score, and it obtains the highest scores in all three metrics. The overall accuracy comparison is shown in **Figure 8.b.** All model configurations achieve exceptionally high accuracy, exceeding 98.8%, according to the results. The model that uses all 46 of the CICIoT2023 dataset's features achieves the highest accuracy of 0.9912. The TON-IoT dataset, which has 43 features and an accuracy of 0.9908, comes in close second. Furthermore, the training and validation accuracy with loss function of model on the TON_IoT shown in **Figure 9**.

The model's high effectiveness is confirmed by validation across several datasets. Strong results with a smaller feature set also demonstrate the possibility of creating models that are more computationally efficient without reducing performance significantly.



(a) Precision, Recall and F1-Score Mertics



(a) Model Accuracy

Figure 8 (a,b). Comparison model performance metrics in both CICIoT2023 vs TON_IoT dataset

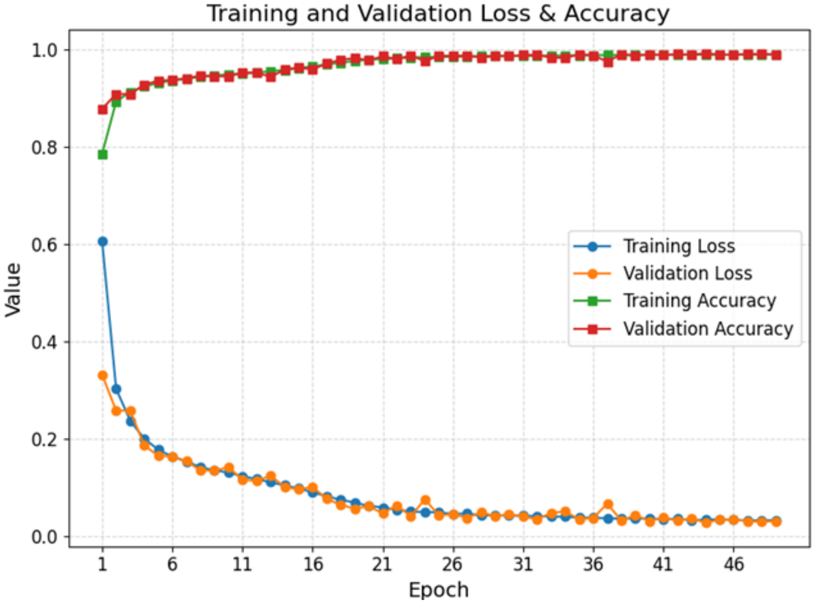


Figure 9. Model Accurcy and Loss Function on TON_IoT

5.6 Performance Analysis of Features Selection Methods

To validate the efficiency of our technique for feature selection, SHAP was compared to three popular feature selection techniques: Recursive Feature Elimination (RFE), Principal Component Analysis (PCA), and Mutual Information (MI). We chose the top ten features for each approach and used the same structure to train our CNN-LSTM model. With an accuracy of 99.00% and a macro F1-score of 98.13%, SHAP outperformed RFE (97.16%), MI (95.71%), and PCA (95.67%), as shown in **Figure 10**. Despite providing precise feature importance ranking, SHAP's computational complexity might be too high for real-time use on IoT devices with limited resources. As a result, we only employ SHAP offline for feature selection and model training. Without performing any additional SHAP calculations, the lightweight model is deployed after the most pertinent features have been identified. This benchmarking validates SHAP's suitability for our IoT anomaly detection framework by confirming that it not only offers model interpretability but also superior predictive performance.

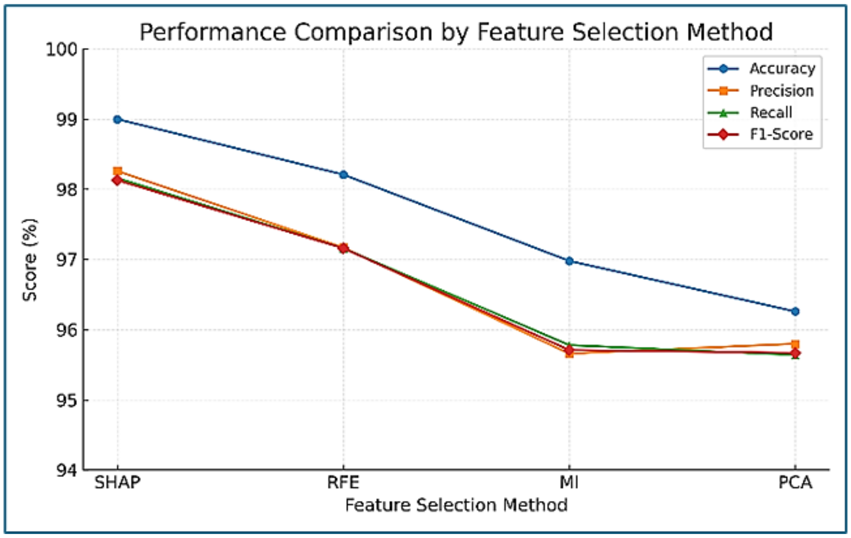


Figure 10. Comparing CNN-LSTM performance (accuracy, precision, recall, and F1-score) across four feature-selection techniques (SHAP, RFE, Mutual Information, and PCA)

6. Discussion

This research proposes a lightweight and explainable anomaly detection framework designed for IoT environments, combining a hybrid CNN-LSTM architecture with SHAP-based feature selection and SMOTE-ENN balancing. The proposed framework was validated against the CICIoT2023 dataset, showing encouraging performance and efficiency results while addressing major challenges faced by real-time intrusion capture systems. The original model, trained with all 46 features, therefore achieved an accuracy of 99.12% across 14 attack classes, demonstrating its capability in multiclass classification. However, training with the full feature set resulted in computational costs and redundancy. In order to resolve this top 10 most impactful features were selected using SHAP leading to a retrained CNN-LSTM model that maintained a high accuracy of 99.00% while improving computational efficiency making the calculations much faster. Inference latency on the GPU decreased from 0.000082 seconds per sample to 0.000067 seconds, and on the CPU from 0.000182 seconds to 0.000084. This demonstrates how SHAP can enhance model interpretability while also optimizing resources. Furthermore, we evaluated SHAP against three well-known feature selection techniques, and SHAP performed better than each of them in terms of choosing the most instructive features. Class imbalance, a common limitation in cybersecurity datasets, was addressed using SMOTE-ENN, which proved effective in improving detection performance across minority attack classes also recall and F1-score improvements were observed across these challenging categories, demonstrating the importance of effective balancing techniques for robust detection. The proposed model also tested for generalizability on the ToN-IoT dataset, where it demonstrated robustness across various datasets with an exceptional accuracy of 99.08%. This shows that the model can work with different types of devices and traffic patterns in different IoT environments.

To further position our framework within the existing research landscape, **Table 10** presents a comparative overview of recent deep learning models applied to IoT intrusion detection (Gueriani et al., 2024) applied a CNN-LSTM model on CICIoT2023, achieving 98% accuracy using 825,886 parameters, but only for binary classification highlighting a limited problem scope and significantly higher complexity than our approach. Similarly,(Hizal et al., 2024) utilized a CNN-only model with 8.89 million parameters to classify 12 attack categories, yielding 91% accuracy demonstrating inefficiency in model size and performance trade-offs. In contrast, our model achieves 99.12% accuracy across 14 classes with just 43,406 trainable parameters, offering a substantially more compact and scalable solution (Alzahrani et al., 2024) proposed a smaller CNN-LSTM model with 3,558 parameters, reaching 92.2% accuracy on 6 classes, whereas our model handles more than twice the number of classes with significantly higher accuracy (Nazir et al., 2024), applying CNN-LSTM to CICIDS2017, attained 92% accuracy across 15 classes with 3,848 parameters, indicating strong efficiency but limited detection performance (Wang et al., 2023a) used CNN-LSTM on CICIDS2018, achieving 98.8% accuracy with 2.79 million parameters a much more resource-intensive approach.(Wang et al., 2023b) applied DL-BiLSTM and CNN models separately on CICIoT2023, achieving 93% and 91% accuracy on 8 classes, using 1,982 and 42,952 parameters, respectively, this comparative analysis underscores the strength of our approach in balancing model performance, classification granularity, and computational complexity, therefor integrating SHAP for explainable and efficient feature selection and using SMOTE-ENN for enhanced data balance, the proposed CNN-LSTM framework demonstrates a practical and scalable solution for real-time IoT intrusion detection. It offers broader class coverage, significantly reduced parameter count, and high accuracy, making it well-suited for deployment in resource-constrained environments where low latency and high reliability are essential. Despite the model is lightweight (~43K parameters), implementing deep learning models in resource-constrained devices requires further optimization like pruning and quantization. Incorporating continuous learning mechanisms could further enhance adaptability, enabling the framework to stay effective in the face of evolving threats.

Table 10. Comparison of Deep Learning Models on IoT Intrusion Detection Datasets

Ref	Year	Model	Dataset	Accuracy	Parameters	Classification	Features
(Nazir et al., 2024)	2024	Cnn-lstm	Cicids2017	92%	3,848	15 classes	All
(Gueriani et al., 2024)	2024	Cnn-lstm	Ciciot2023	98%	825,886	2 classes	46
(Hizal et al., 2024)	2024	cnn	Ciciot2023	91%	8,889,612	12 classes	46
(Wang et al., 2023a)	2023	Cnn-lstm	Cicids2018	98.8%	2,793,767	7 classes	70
(Wang et al., 2023b)	2023	DL-BiLSTM , Cnn	Ciciot2023	93%, 91%	1982 , 42952	8 classes	46
(Alzahrani et al., 2024)	2024	Cnn-lstm	Ciciot2023	92.2%	3,558	6 classes	46
Proposed	2025	Cnn-lstm	Ciciot2023	99.12	**43,406**	14 classes	46
Proposed	2025	Cnn-lstm	Ciciot2023	99.00	**43,406**	14 classes	10

7.Conclusion and Future Work

In this framework, we propose a resilient, interpretable, and lightweight anomaly detection framework for IoT security. The model addresses key challenges such as excessive high dimensional feature redundancy, class imbalance, and low interpretability by combining a hybrid CNN-LSTM architecture with SHAP-based feature selection and SMOTE-ENN balancing. It outperforms existing deep learning-based intrusion detection systems in both accuracy and computational efficiency, achieving 99.12% accuracy across 14 classes with only 43,406 trainable parameters. While the evaluation was conducted in the controlled Kaggle environment, the results strongly support the effectiveness of the model. Future work should prioritize the facility of deploying and validating the model in real-time, resource-constrained edge computing scenarios to assess its responsiveness and resilience under operational constraints. This will be crucial for smart city, industrial IoT, and real-time monitoring use cases. To evaluate the model's generalizability under various network conditions and attack patterns, we also tested it on the ToN-IoT dataset. This confirmed the model's strong performance across a variety of dynamic IoT environments. Incorporating continuous learning mechanisms could further enhance adaptability, enabling the system to stay effective in the face of evolving threats. Real-time threat detection on resource-constrained edge devices, strong protection for vital infrastructure like power grids and transportation networks, and improved security for healthcare IoT systems are all anticipated benefits of the suggested anomaly detection framework. Furthermore, it helps lower operating costs through effective, on-device analysis that reduces dependency on cloud resources and promotes smart city resilience by averting significant service interruptions. The findings of this research establish a solid foundation for advancing intelligent, interpretable, and scalable intrusion detection systems for modern IoT environments. The balance between accuracy, efficiency, and transparency makes this approach well-suited for practical cybersecurity deployments.

References

Albulayhi, K., Al-Haija, Q. A., Alsuhibany, S. A., Jillepalli, A. A., Ashrafuzzaman, M. & Sheldon, F. T. 2022. IoT Intrusion Detection Using Machine Learning with a Novel High Performing Feature Selection Method. *Applied Sciences (Switzerland),* 12.

Alzahrani, H., Sheltami, T., Barnawi, A., Imam, M. & Yaser, A. 2024. A Lightweight Intrusion Detection System Using Convolutional Neural Network and Long Short-Term Memory in Fog Computing. *Computers, Materials and Continua,* 80**,** 4703-4728.

Anwer, M. A., Qattan, G. A. & Ali, A. M. 2024. Ocular disease classification using different kinds of machine learning algorithms. *Zanco Journal of Pure and Applied Sciences,* 36**,** 25-34.

Bakhsh, S. A., Khan, M. A., Ahmed, F., Alshehri, M. S., Ali, H. & Ahmad, J. 2023. Enhancing IoT network security through deep learning-powered Intrusion Detection System. *Internet of Things (Netherlands),* 24.

Gueriani, A., Kheddar, H. & Mazari, A. C. 2024. Enhancing IoT Security with CNN and LSTM-Based Intrusion Detection Systems.

Hajjouz, A. & Avksentieva, E. 2024. Optimizing Intrusion Detection for DoS, DDoS, and Mirai Attacks Subtypes Using Hierarchical Feature Selection and CatBoost on the CICIoT2023 Dataset. *Data and Metadata,* 3.

Hassen, S. & Abdlrazaq, A. 2024. Contextual Deep Semantic Feature Driven Multi-Types Network Intrusion Detection System for IoT-Edge Networks. *Zanco Journal of Pure and Applied Sciences,* 36**,** 132-147.

Hizal, S., Cavusoglu, U. & Akgun, D. 2024. A novel deep learning-based intrusion detection system for IoT DDoS security. *Internet of Things (Netherlands),* 28.

Ji, R., Kumar, N. & Padha, D. 2024. Hybrid Enhanced Intrusion Detection Frameworks for Cyber-Physical Systems via Optimal Features Selection. *Article in Indian Journal of Science and Technology,* 17**,** 3069-3069.

Khan, M. M. & Alkhathami, M. 2024. Anomaly detection in IoT-based healthcare: machine learning for enhanced security. *Scientific Reports,* 14.

Khanday, S. A., Fatima, H. & Rakesh, N. 2024. A Novel Data Preprocessing Model for Lightweight Sensory IoT Intrusion Detection. *International Journal of Mathematical, Engineering and Management Sciences,* 9**,** 188-204.

Krzysztoń, E., Rojek, I. & Mikołajewski, D. 2024. A Comparative Analysis of Anomaly Detection Methods in IoT Networks: An Experimental Study. *Applied Sciences (Switzerland).* Multidisciplinary Digital Publishing Institute (MDPI).

Manokaran, J. & Vairavel, G. 2024. DL-ADS: Improved Grey Wolf Optimization Enabled AE-LSTM Technique for Efficient Network Anomaly Detection in Internet of Thing Edge Computing. *IEEE Access,* 12**,** 75983-76002.

Modi, P. Towards Efficient Machine Learning Method for IoT DDoS Attack Detection.

Moustafa, N. 2021. A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets. 72.

Nazir, A., He, J., Zhu, N., Qureshi, S. S., Qureshi, S. U., Ullah, F., Wajahat, A. & Pathan, M. S. 2024. A deep learning-based novel hybrid CNN-LSTM architecture for efficient detection of threats in the IoT ecosystem. *Ain Shams Engineering Journal,* 15.

Neto, E. C. P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R. & Ghorbani, A. A. 2023. CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment. *Sensors,* 23.

Sanju, P. 2023. Enhancing intrusion detection in IoT systems: A hybrid metaheuristics-deep learning approach with ensemble of recurrent neural networks. *Journal of Engineering Research (Kuwait),* 11**,** 356-361.

Shareef, S. M. 2023. The adoption of the Internet of Things in E-government towards the Smart Government. *Zanco Journal of Pure and Applied Sciences,* 35**,** 67-78.

Shtayat, M. B. M., Hasan, M. K., Sulaiman, R., Islam, S. & Khan, A. U. R. 2023. An Explainable Ensemble Deep Learning Approach for Intrusion Detection in Industrial Internet of Things. *IEEE Access,* 11**,** 115047-115061.

Tabassoum, N., Bindu, F., Sheikh, S., Rab, R., Leshob, A. & Wahab, T. B. Multiclass Feature Selection Model for Adversarial Attacks in IoT Environment. Proceedings -

2024 IEEE International Conference on e-Business Engineering, ICEBE 2024, 2024. Institute of Electrical and Electronics Engineers Inc., 53-59.

Wang, Y. C., Houng, Y. C., Chen, H. X. & Tseng, S. M. 2023a. Network Anomaly Intrusion Detection Based on Deep Learning Approach. *Sensors,* 23.

Wang, Z., Chen, H., Yang, S., Luo, X., Li, D. & Wang, J. 2023b. A lightweight intrusion detection method for IoT based on deep learning and dynamic quantization. *PeerJ Computer Science,* 9.