


CBM-IDS: An Advanced Hybrid Deep Learning Model for DDoS Attack Detection in IoT Networks


Hamdullah Karamollaoglu

(Ministry of Energy and Natural Resources, Electricity Generation Corporation, Ankara,
Türkiye

 <https://orcid.org/0000-0001-6419-2249>, h.karamollaoglu@euas.gov.tr)


İbrahim Yücedağ

(Department of Computer Engineering, Faculty of Engineering, Düzce University, Düzce,
Türkiye

 <https://orcid.org/0000-0003-2975-7392>, ibrahimyucedag@duzce.edu.tr)


İbrahim Alper Dođru

(Department of Computer Engineering, Faculty of Technology, Gazi University, Ankara,
Türkiye

 <https://orcid.org/0000-0001-9324-7157>, iadogru@gazi.edu.tr)


Sinan Toklu

(Department of Computer Engineering, Faculty of Technology, Gazi University, Ankara,
Türkiye

 <https://orcid.org/0000-0002-8147-9089>, stoklu@gazi.edu.tr)

İsmail Atacak

(Department of Computer Engineering, Faculty of Technology, Gazi University, Ankara,
Türkiye

 <https://orcid.org/0000-0002-6357-0073>, iatacak@gazi.edu.tr)

Abstract: The rapid expansion of IoT devices has transformed industries while simultaneously introducing critical security vulnerabilities, particularly Distributed Denial-of-Service (DDoS) attacks that exploit the constrained resources of IoT systems. To address this challenge, a novel intrusion detection system (CBM-IDS) is proposed for the effective identification and mitigation of DDoS attacks in IoT environments. A hybrid deep learning framework is employed, integrating Convolutional Neural Networks (CNN) for spatial feature extraction, Bidirectional Long Short-Term Memory (BiLSTM) for temporal dependency analysis, and a Multi-Head Attention Mechanism (MHAM) to prioritize critical network traffic patterns. Model robustness is enhanced through Adaptive Synthetic Sampling (ADASYN) and One-Sided Selection (OSS) for class imbalance mitigation, along with dimensionality reduction using an Autoencoder combined with ANOVA F-test-based feature selection. The proposed system is evaluated on the CICDDoS2019 benchmark dataset, achieving a detection accuracy of 99.93%, which demonstrates its efficacy in real-world IoT security applications.

Keywords: Intrusion detection, IoT network, deep learning, feature engineering, DDoS attacks

Categories: H.3.1, H.3.2, H.3.3, H.3.7, H.5.1

DOI: 10.3897/jucs.146099

1 Introduction

Despite the numerous advantages of IoT in daily life, its widespread deployment has introduced significant security vulnerabilities. Most devices used in IoT applications are particularly vulnerable to cyber-attacks owing to their limited processing power and bandwidth [Alwahedi et al., 24; Chaabouni et al., 19]. DDoS attacks are among the most common and disruptive forms of cyber threats. They aim to flood the target system or network with malicious traffic, rendering it inaccessible to legitimate users [Salim, 20; Saiyed, 24]. With the increasing prevalence of such attacks, intrusion detection systems (IDSs) have become critical, as traditional mechanisms like firewalls and anti-virus software are insufficient for detecting complex DDoS behaviors [Najafimehr et al., 22]. Although IDSs can be categorized based on their location or detection method, detailed distinctions such as Host-based and Network-based architectures have been minimized in this study to maintain focus on the primary contribution. Likewise, general explanations regarding signature-based and anomaly-based detection approaches have been condensed, as the intended audience already understands their roles well [Martins et al., 22; Otoum and Nayak, 21].

Despite ongoing developments, traditional IDS implementations face persistent challenges. Many rely on datasets skewed toward benign traffic, making it difficult to detect emerging or rare attack types. Additionally, high-dimensional feature spaces increase computational demands and negatively impact accuracy. Traditional machine learning (ML) methods, while efficient in training, often exhibit limited effectiveness under such constraints. Their performance typically degrades when handling imbalanced datasets or complex attack patterns. In contrast, deep learning (DL) models are capable of extracting hierarchical and non-linear representations, enabling the detection of sophisticated threats that traditional models may miss [Thapa et al., 20; Ahmad et al., 21]. However, these benefits come with challenges, including increased computational cost and longer training durations.

To mitigate these challenges, this study proposes CBM-IDS, which integrates advanced data balancing, feature engineering techniques, and DL architectures. CBM-IDS employs autoencoder and the analysis of variance (ANOVA) F-Test for dimensionality reduction, effectively reducing the feature space's complexity while preserving essential information. To address the issue of dataset imbalance, techniques such as the Adaptive Synthetic Sampling (ADASYN) and the One-Sided Selection (OSS) methods are employed, aiming to achieve a balanced representation of both normal and malicious instances. Additionally, a novel hybrid DL model is used for classification. This innovative hybrid DL model integrates Conv1D layers to identify spatial patterns within one-dimensional sequences and BiLSTM layers to capture bidirectional sequential dependencies. By combining these components, the model gains the ability to effectively capture intricate patterns and relationships inherent in the data, contributing to improved classification accuracy. Moreover, by incorporating the multi-head attention mechanism (MHAM) before the dense layer, the model enhances its capability to focus on relevant information and handle long-range dependencies effectively. By attending to different parts of the input sequence simultaneously, MHAM enhances the model's ability to extract meaningful features, ultimately leading to improved classification performance. By combining these components with advanced data balancing and dimensionality reduction techniques,

CBM-IDS aims to enhance detection accuracy while maintaining efficiency in resource-constrained IoT environments. The main contributions of this study can be summarized as follows:

1. A novel IDS, termed CBM-IDS, is proposed for the identification of DDoS attacks in IoT environments. In contrast to existing approaches, CBM-IDS incorporates a unique integration of dimensionality reduction, data balancing, and hybrid deep learning techniques.
2. Advanced data handling techniques are integrated, including the utilization of autoencoder and ANOVA F-test for dimensionality reduction to enhance feature extraction capabilities crucial for identifying DDoS patterns. Additionally, ADASYN and OSS methods are implemented to mitigate data imbalance issues by generating synthetic samples and optimizing minority class representation.
3. The study presents a hybrid DL model that integrates CNN for extracting spatial features from IoT data streams, BiLSTM for capturing temporal dependencies and sequential patterns, and MHAM to enhance the model's focus on relevant information, thereby improving classification accuracy for DDoS detection.
4. K-fold cross-validation (CV) is employed to ensure unbiased performance evaluation of the CBM-IDS framework, providing reliable estimates of its effectiveness in detecting DDoS attacks.
5. Hyperparameter tuning was conducted using the HyperOpt library with the Tree-structured Parzen Estimator (TPE) algorithm to optimize both the Autoencoder-ANOVA F-Test feature reduction pipeline and the CNN-BiLSTM-MHAM classification model.
6. The implementation includes “EarlyStopping” and “ReduceLRonPlateau” callbacks to prevent overfitting and optimize the training process.

The rest of the paper is organized as follows: Section 2 reviews related studies on intrusion detection in both IoT and other computer networks. Section 3 outlines the materials and methods, including datasets, preprocessing, feature engineering, data balancing, and the hybrid DL model. Section 4 presents the experimental setup and results analysis. Section 5 concludes the paper, summarizing key findings and future research directions.

2 Related Works

In recent years, there has been a marked increase in research aimed at developing IDSs to identify and prevent attacks on computer networks. This review summarizes recent work on IDS approaches developed for detecting attacks on various networks, such as IoT, fog, and traditional networks. In addition to IDS-specific studies, recent research has also focused on enhancing network security and efficiency under dynamic and constrained conditions. Approaches such as reputation-based routing protocols have been proposed to mitigate the effects of misbehaving nodes in mobile ad hoc networks [Mohajer et al., 13a]. Traffic-aware network slicing and adaptive offloading strategies have been developed using reinforcement learning to optimize resource allocation and service quality in mobile edge computing environments [Mohajer et al., 24]. Moreover, network coding combined with power control techniques has been applied to enable

secure and cost-effective multicast communication in interference-limited dynamic networks [Mohajer et al., 13b]. These complementary advancements at the network level contribute significantly to the robustness and adaptability of modern intrusion detection frameworks.

Feature reduction combined with classification has been widely adopted to improve the performance of IDSs by addressing high dimensionality and enhancing classification accuracy. Several studies propose hybrid models that leverage complementary techniques for feature extraction and classification. Zheng et al. [Zheng et al., 20] developed a hybrid IDS using Linear Discriminant Analysis (LDA) for feature reduction and Extreme Learning Machine (ELM) for classification, achieving an accuracy rate of 92.35% on the NSL-KDD dataset. de Souza et al. [de Souza et al., 20] proposed a hybrid model using Information Gain (IG) for feature selection and a hybrid DNN-K-Nearest Neighbors (KNN) approach for classification, obtaining impressive accuracy rates of 99.77% on the NSL-KDD dataset and 99.85% on the CIC-IDS2017 dataset. Kunang et al. [Kunang et al., 21] introduced a hybrid IDS that combined Denoising Autoencoder (DAE) and Deep Neural Network (DNN) methods for feature extraction and classification. They achieved notable accuracy rates of 83.33% on the NSL-KDD dataset and 95.97% on the CIC-IDS2018 dataset. Laghrissi et al. [Laghrissi et al., 21] combined PCA and MI methods for dimensionality reduction in their developed IDS. They employed LSTM for classification, and the system was evaluated on the KDDCup'99 dataset, resulting in an accuracy of 99.44%. Labiod et al. [Labiod et al., 22] developed a lightweight IDS using Variational Autoencoder (VAE) for feature extraction and Multi-Layer Perceptron (MLP) for classification, achieving high accuracy rates of 99.98% on the IoT-23 dataset and 98.98% on the IoTID20 dataset.

ML and DL classifiers have been extensively used to detect complex intrusion patterns, often combined with feature selection or data balancing techniques to enhance performance. Gökdemir et al. [Gökdemir et al., 22] implemented a lightweight anomaly detection approach using SVM, Naive Bayes (NB), and Long Short-Term Memory (LSTM) methods, achieving accuracy rates of 98.16% with SVM, 84.52% with NB, and 99.17% with LSTM on the DAD dataset. Keserwani et al. [Keserwani et al., 21] introduced an IDS for IoT networks using Grey Wolf Optimizer (GWO) and Particle Swarm Optimization (PSO) for feature selection, and RF for classification, obtaining accuracy rates of 99.66% on the KDDCup'99 dataset, 99.24% on the NSL-KDD dataset, and 99.88% on the CICIDS-2017 dataset. Cao et al. [Cao et al., 22] proposed a hybrid IDS combining RF and Pearson's Correlation Coefficient (PCC) for feature selection, ADASYN, and Random Oversampling with Replacement (RENN) for data balancing, and CNN-Gated Recurrent Unit (CNN-GRU) for classification. They achieved accuracy rates of 86.25% on the UNSW-NB15 dataset, 99.69% on the NSL-KDD dataset, and 99.65% on the CIC-IDS2017 dataset. Saba et al. [Saba et al., 22] introduced a CNN-based IDS for the IoT environment on the NID and BoT-IoT datasets, achieving accuracies of 99.51% and 95.55%, respectively. Samha et al. [Samha et al., 23] employed a hybrid CNN and Deep Watershed Autoencoder (CNN-DWA) in their developed IDS. They conducted testing using the KDD-Cup'99 dataset and achieved an accuracy of 98.05%. Du et al. [Du et al., 23] utilized a hybrid model combining CNN and LSTM in their developed IDS. They conducted testing on the KDD CUP'99 dataset, achieving 97% accuracy; on the NSL-KDD dataset, reaching 99% accuracy; and on the UNSW-NB15 dataset, obtaining 94% accuracy. Bacha et al.

[Bacha et al., 24] introduced an IDS framework that combines Kernel Principal Component Analysis (KPCA) for feature reduction with Kernel ELM for classification. They tested this framework on the UNSW-NB15 dataset, achieving an accuracy of 98.64%. Vibhute et al. [Vibhute et al., 24] developed an IDS comprising a feature selection module based on the RF model, coupled with a classification module utilizing LSTM networks. They evaluated their approach on the CSE-CICIDS2018 dataset, achieving an accuracy of 99.66%. Sayegh et al. [Sayegh et al., 24] introduced an IDS tailored for IoT networks. To tackle data imbalance, they implemented the Synthetic Minority Over-sampling Technique (SMOTE). For feature selection, Recursive Feature Elimination (RFE) was employed, and for classification, LSTM was utilized. Their model demonstrated accuracies of 99.34% on the CICIDS2017 dataset, 99.67% on NSL-KDD, and 98.31% on UNSW-NB15.

Ensemble learning approaches, which combine multiple base classifiers, have been successfully applied to improve IDS performance and robustness. Mighan and Kahani [Mighan and Kahani, 21] utilized Stacked Autoencoder (SAE) for feature reduction and Support Vector Machine (SVM) for classification, achieving accuracy rates of 90.2% on the ISCX-2012 dataset and 99.49% on the CICIDS-2017 dataset. Krishnaveni et al. [Krishnaveni et al., 21] employed an ensemble approach consisting of SVM, NB, LR, and DT classifiers, along with Unified Ensemble Feature Selection (UEFFS) for feature selection. Their IDS achieved accuracy rates of 96.06% on the NSL-KDD dataset, 99.93% on the Kyoto 2006+ dataset, and 98.89% on the Real-time Honeypot dataset. Alotaibi et al. [Alotaibi et al., 23] used the TON-IoT dataset and employed four classifiers: RF, Decision Tree (DT), LR, and KNN. These four classifiers were then incorporated into two ensemble approaches: voting and stacking. The ensemble methods achieved an accuracy rate of 98.63%. Cao et al. [Cao et al., 23] introduced an IDS utilizing a stacked ensemble model, merging NB and LightGBM (LGBM) learners. Their study evaluated the IDS on the NB-aIoT and UNSW-NB15 datasets. Findings revealed that the proposed IDS achieved an average accuracy rate of 99.68%. Roopak et al. [Roopak et al., 23] proposed an Unsupervised approach-based IDS for detecting zero-day DDoS attacks on IoT networks. The performance evaluation of the proposed IDS was conducted using the CICDDoS2019 dataset. The Gaussian Random Projection (GRP) method was employed for feature reduction in the developed IDS. For attack classification, an ensemble model combining One-Class SVM, K-Means Clustering, and Gaussian Mixture Model (GMM) was suggested. As a result, an accuracy of 94.55% was achieved in attack classification. Zhu et al. [Zhu et al., 24] utilized three base learners, including NB, LGBM, and XGBoost, along with a Logistic Regression (LR) meta-learner. They employed PSO for hyperparameter tuning. Their framework was evaluated on the UNSW-NB15 dataset, resulting in an accuracy of 97.05%.

Several previous studies have proposed IDS models using ML and DL techniques, often evaluated on outdated datasets. In contrast, this study uses the more recent and realistic CICDDoS2019 dataset, ensuring a more relevant evaluation against current cybersecurity threats. However, many of these studies did not combine feature reduction and data balancing before classification, which may result in high dimensionality, class imbalance, and overfitting. To address these issues, the proposed CBM-IDS model uses a hybrid feature processing strategy that combines an autoencoder for capturing complex feature relationships with the ANOVA F-test for selecting statistically significant features. This approach produces a compact yet informative feature set that improves both model performance and computational

efficiency. Unlike traditional IDS models that rely on lightweight classifiers and often fail to capture spatial and temporal patterns, CBM-IDS incorporates CNN layers to extract spatial features, BiLSTM units to model temporal dependencies, and a multi-head attention mechanism to emphasize critical time steps. Model efficiency is further enhanced through Bayesian-based hyperparameter optimization, which enables automatic selection of optimal configurations and reduces the need for manual tuning. This contributes to faster inference and lower complexity, making the model suitable for real-time intrusion detection in resource-constrained IoT environments. Additionally, a hybrid ADASYN+OSS sampling technique is employed to generate cleaner and more balanced training data, offering advantages over standard oversampling methods such as SMOTE [Chawla et al., 02]. By addressing challenges like data imbalance, high dimensionality, and overfitting, the CBM-IDS model delivers more accurate and reliable detection performance.

3 Material and Method

This section provides a comprehensive overview of the materials and methods used in the study. Initially, the dataset is presented. Following that, CBM-IDS and its components are illustrated through a diagram. Subsequently, the components that make up the model are explained in detail.

3.1 Dataset

In this study, the CIC-DDoS2019 dataset was utilized for training and performance testing of the CBM-IDS. It includes benign samples and a variety of contemporary DDoS attack samples, emulating real-world scenarios. The dataset, collected over two distinct days, provides a comprehensive range of DDoS attacks utilizing TCP/UDP application layer protocols. The taxonomy of attacks is divided into reflection-based and exploitation-based intrusions, encompassing more than 87 flow features and one label. Reflection-based attacks involve sending malicious packets to intermediary servers, with the source IP address set to the victim's IP, overwhelming the victim with response packets. Samples include MSSQL, SSDP, NTP, TFTP, DNS, LDAP, NetBIOS, and SNMP. Exploitation-based attacks exploit specific network, transport, or application layer protocols, typically overwhelming the victim's resources with TCP or UDP packets. Samples include SYN flood, UDP flood, and UDP-Lag. To ensure privacy while preserving data integrity, the dataset anonymizes network traffic without removing payloads. This approach ensures that the dataset remains comprehensive for analyzing network interactions. The dataset is particularly valuable for research due to its complete traffic capture, attack diversity, data source heterogeneity, and realistic background traffic [Sharafaldin et al., 19; Talukder and Uddin, 23]. The CICDDoS2019 dataset is the most up-to-date resource for studying DDoS attacks, incorporating modern reflective DDoS attacks not captured in earlier datasets. Its inclusion of numerous samples covering current attack behaviors makes it well-suited for developing and evaluating automated IDSs using ML or DL models. Moreover, its extensive feature set serves as a robust foundation for research in network security and DDoS mitigation efforts.

3.2 Configuration of CBM-IDS

CBM-IDS in this study consists of three main phases: The preprocessing and data splitting phase, the dimensionality reduction and data balancing phase, and the classification phase. The workflow of CBM-IDS is illustrated in Figure 1.

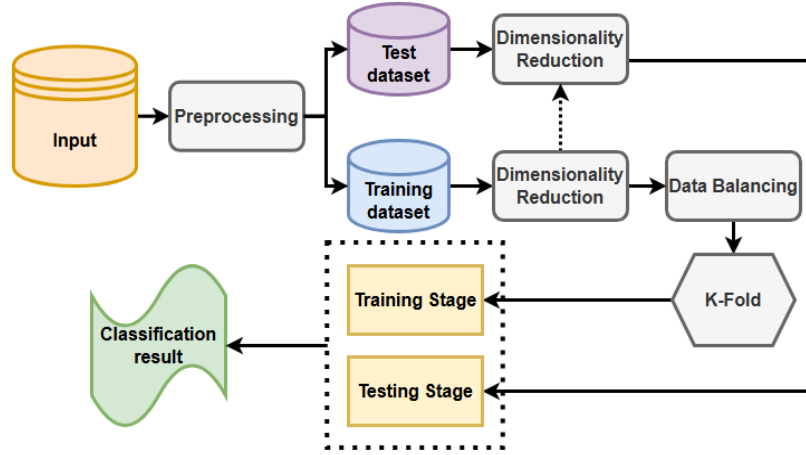


Figure 1: Workflow of CBM-IDS

As seen in the Figure 1, in the first phase, various preprocessing tasks are performed on the dataset to prepare it for further processing. These tasks include eliminating null, duplicate, and infinite values, numericalization using the LabelEncoder from the scikit-learn (sklearn) library [Varoquaux et al., 15], normalization using the MinMaxScaler from the same library, and removing constant features and socket-based features that do not contribute to classification or may lead to overfitting. The second phase involves dimensionality reduction and data balancing. Dimensionality reduction techniques such as autoencoder and ANOVA F-test are utilized to reduce the number of features while retaining relevant information. Data balancing techniques like ADASYN and OSS are employed to address class imbalance issues. The combined utilization of these techniques enhances the effectiveness of subsequent DL models for classification. Finally, in the third phase, classification tasks are carried out using a hybrid method that combines CNN, BiLSTM, and MHAM components. CNNs use convolutional layers to discern spatial patterns or features within data proficiently. In the domain of intrusion detection, this capability enables CNNs to effectively detect intricate irregularities in network traffic data, enhancing the identification and mitigation of potential security threats. The BiLSTM component is employed for sequential information processing. It processes data in both forward and backward directions, capturing dependencies from both the past and the future. This bidirectional processing is essential for analyzing sequential data like network traffic, where the sequence of events is crucial for understanding the context of potential intrusions. Before the data reaches the dense layer for classification, the MHAM component, a multi-head attention mechanism, is utilized. This mechanism allows various “heads” to focus on

different parts of the input, modeling interactions between different features. Each head provides a unique perspective or feature scale, enabling the capture of more complex relationships among features. This step enhances the feature representation, preparing well-refined inputs for the dense layer to achieve accurate classification.

3.3 Preprocessing and Data Splitting Phase

Handling missing, infinite, or duplicate values is a crucial step in data preprocessing to ensure the quality and integrity of the datasets. Null values can introduce errors in statistical calculations, reduce the sample size, and impact the distribution of data. Similarly, duplicate values can distort the accuracy of the analysis and introduce bias in the final results. Additionally, infinite values can lead to Not a Number (NaN) values, significantly affecting the accuracy of mathematical operations, such as division by zero or logarithmic functions. Hence, in the data preprocessing stage, the removal of irrelevant or noisy data was prioritized in the datasets utilized for this study. After eliminating null, duplicate, and infinite values, further preprocessing steps were carried out. Data encoding was performed using Label Encoder from the sklearn library to convert categorical data into a numerical format. Data normalization was then implemented using MinMaxScaler from the same library. This step ensures that all features contribute equally to the analysis and prevents variations or unexpected results by scaling them to a specified range, typically between 0 and 1. Following the preprocessing steps, constant features, which maintain the same value across all instances within the dataset, were removed to enhance the dataset's quality. Their removal is crucial to prevent overfitting during the training phase, as these features consistently maintain the same value across the vast majority of instances. For instance, features like "Bwd PSH Flags" and "Fwd URG Flags" have a value of 0 for all instances, representing 100% of the dataset. Similarly, the "SYN Flag Count" feature has a value of 0 for 99.95% of the dataset, making it quasi-constant. By eliminating these features, the training dataset contains only relevant and informative features, thereby improving the overall performance of CBM-IDS.

The original dataset consisted of 88 features. During the preprocessing phase, 10 socket-related features ("Unnamed:0," "Flow ID," "Source IP," "Source Port," "Destination IP," "Destination Port," "Protocol," "Timestamp," "Inbound," and "Similar HTTP") [Wei et al., 21; Batchu and Seetha, 22; Ahmim et al., 23] were identified as irrelevant to the training process and consequently removed from the dataset because they do not contribute to detecting DDoS attacks and may lead to overfitting issues due to the model's bias towards them. As a result, the dataset was reduced to contain 64 features and 1 label for effective model training and testing. Subsequently, the dataset instances were divided into two subsets: a training set containing 80% of the data and a testing set containing the remaining 20%. This division was achieved using the "train_test_split" function from the sklearn library. This approach ensures that the model can be trained on a substantial portion of the data while retaining a separate set to evaluate its performance. Upon completing the preprocessing and data splitting steps, the class distributions of the CICDDoS2019 dataset are summarized in Table 1. This table provides an overview of the number of instances and their corresponding percentages across various classes in both training and testing subsets.

Class	Training	Testing	Total	Percentage
Benign	78212	19619	97831	22.68 %
Attack	266884	66656	333540	77.32 %
Total	345096	86275	431371	100 %

Table 1: Distribution of training and testing data for benign and attack classes

As seen in Table 1, there is a significant imbalance between the benign and attack classes, with the Attack class comprising approximately 3.41 times more instances than the Benign class.

3.4 Dimensionality Reduction and Data Balancing Phase

In this phase, the integration of dimensionality reduction and data balancing techniques was utilized to enhance the training process and the model's classification performance. Dimensionality reduction aims to reduce the number of input features while retaining essential information, thereby improving computational efficiency and model accuracy. Data balancing addresses class imbalance in datasets, ensuring a fair representation of all classes during the training process and enhancing model robustness.

In this study, dimensionality reduction was applied to the training data. The transformation obtained from the training data was subsequently applied to the test data, projecting it into the same reduced feature space. Dimensionality reduction is crucial for mitigating the effects of sparse data and reducing training time, consequently improving the model accuracy. By reducing the number of input features, redundant and irrelevant data are eliminated, thus enhancing the model's efficiency [Xiao et al., 19]. To achieve this, a hybrid approach combining autoencoder and the ANOVA F-Test method was employed for dimensionality reduction. Initially, autoencoder, an unsupervised neural network technique, were utilized to compress the data. Autoencoder effectively capture complex patterns and reduce noise, resulting in a condensed version of the dataset that retains essential information [Wang et al., 22]. However, autoencoder alone do not identify which features are most relevant for the prediction task. To address this limitation, the ANOVA F-Test was applied following the use of autoencoder. The ANOVA F-Test is a statistical method that evaluates the significance of each feature by comparing the variance within groups to the variance between groups [Bommert et al., 20]. By applying the ANOVA F-Test to the data reduced by the autoencoder, the most relevant features from the compressed dataset were selected. This two-step process combines the strengths of both methods. The autoencoder reduces dimensionality by learning intricate feature representations, while the ANOVA F-Test ensures that the selected features are highly relevant to the prediction task. This hybrid approach enhances model efficiency and improves overall performance.

In this study, hyperparameter optimization of the autoencoder model was performed using the HyperOpt library [Bergstra et al., 2013], with the TPE employed as the search algorithm. The optimization process aimed to maximize classification accuracy, which was designated as the performance metric to evaluate the model's effectiveness. The search space for each hyperparameter, along with the optimal values selected through this process, is described below. The dimensionality reduction

parameter (`n_dim_red`) was explored within an integer range from 1 to the original feature dimension, with the optimal value determined as 25. Among the evaluated optimizers Adam, RMSprop and SGD, RMSprop demonstrated the best performance and was therefore selected. The learning rate was tuned over the range 0.001, 0.01, and 0.1, with 0.001 identified as the most effective value. The model training process was configured to run for 40 epochs, selected from the evaluated options [10, 20, 40]. For the activation function, hyperbolic tangent (`tanh`) was preferred over Rectified Linear Unit (ReLU) and softmax and was consistently used in both encoder and decoder layers. The loss function was selected as `mean_squared_error`, based on comparisons with `logcosh` and `binary_crossentropy`. In terms of hidden layer architecture, the configuration [64, 32] (comprising two hidden layers) provided the best performance among the alternatives. A dropout rate of 0.2 was applied to prevent overfitting, and the batch size was set to 64, which was optimal among the options of 32, 64, and 128. Batch normalization was enabled to stabilize and accelerate the learning process. The encoder and decoder components were both assigned a dropout rate of 0.1, contributing to generalization. Regularization was also incorporated into the model: the L1 regularization parameter was set to 0.001, and L2 was set to 0.01, selected from their respective candidate values of [0.001, 0.01, 0.1]. For kernel initialization, the `he_normal` method was found to be most effective compared to `random_normal` and `glorot_uniform`. To validate the model during training, 20% of the training dataset was reserved as a validation set. Additionally, data shuffling was enabled (`shuffle=True`) to ensure randomness in batch selection and promote generalization. This comprehensive optimization strategy led to a robust autoencoder configuration with strong performance on the target task.

The feature importance scores for the 25 features obtained through the ANOVA F-Test method following the autoencoder reduction are presented in Figure 2.

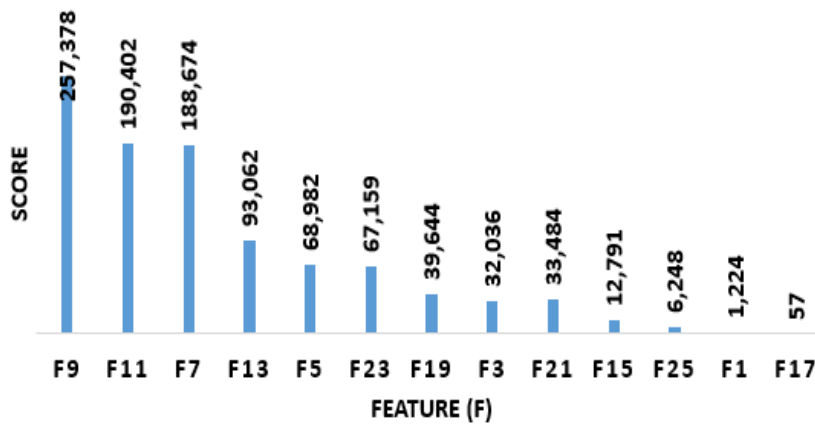


Figure 2: The feature importance scores obtained using the ANOVA F-Test method

As seen in Figure 2, after applying the ANOVA F-Test method following the autoencoder reduction, the highest score is achieved by F8, reaching 257378. Conversely, F16 has the lowest score of 57. To address the data balancing challenge, the study employs a hybrid method that integrates the ADASYN with OSS. ADASYN

stands out from similar methods due to its adaptive approach in synthetic sample generation. Instead of simply increasing all minority class samples uniformly, ADASYN focuses on areas within the minority class that are more complex to learn [Liu et al., 21]. ADASYN enhances the minority class representation by generating synthetic samples based on the local density and learning difficulty of instances, ensuring that synthetic samples are created where they are most needed. This mitigates the imbalance issue by improving the overall balance of the dataset [He et al., 08]. After applying the ADASYN, the OSS is essential for cleaning the dataset by selectively removing noisy instances from the majority class that overlap with the minority class samples. The OSS operates by detecting and eliminating majority-class instances near minority-class samples [Kubat and Matwin, 97]. This approach helps to clarify the boundaries between classes and enhances classification accuracy. This process ensures that the synthetic samples generated by the ADASYN accurately reflect the proper distribution of the minority class without being distorted by irrelevant majority-class data. This two-step process combines ADASYN’s method of boosting minority class representation with OSS’s methodology of cleaning the dataset. By ensuring a balanced representation and removing noisy data points, this approach enhances the dataset’s quality for training classification models on imbalanced datasets. Figure 3 shows the distribution of samples in the training dataset before data balancing, after applying the ADASYN method, and after using ADASYN combined with OSS.

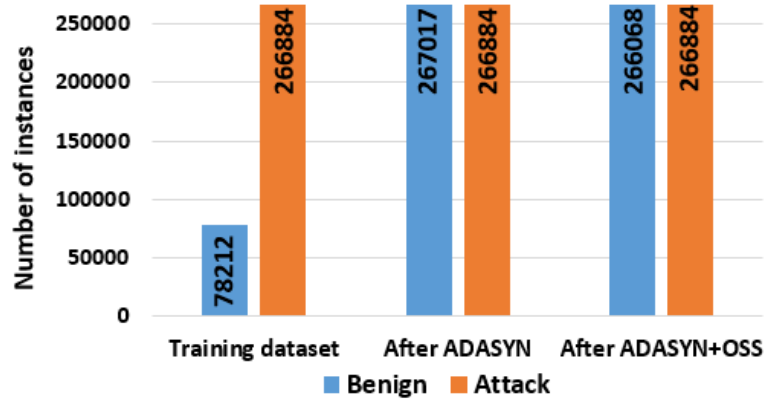


Figure 3: Sample distribution in the training dataset before and after data balancing

As shown in Figure 3, following the combined use of ADASYN and OSS for data balancing, the number of samples for the minority class, Benign, has increased from 78212 to 266068. In contrast, the number of samples for the majority class, Attack, remains constant at 266884. Consequently, the total number of samples in the training dataset has risen from 345096 to 532952 after the data balancing process, representing an approximate 54.4% increase. By integrating ADASYN and OSS, this hybrid framework leverages the complementary strengths of ADASYN’s local density-based approach and OSS’s selective undersampling technique, resulting in improved performance and robustness in addressing class imbalance challenges.

3.5 Classification Phase Using the Proposed Hybrid Deep Learning Approach

In the rapidly evolving field of network security, traditional methods often fall short in detecting sophisticated and evolving cyber threats [Ozkan-Ozay et al., 24]. To address these challenges, a hybrid DL approach is proposed in this study, combining the strengths of CNN, BiLSTM, and MHAM methods. This hybrid model is designed to leverage the spatial feature extraction capabilities of CNNs, the temporal dependency capturing ability of BiLSTMs, and the powerful sequence modeling features of MHAM. By integrating these models, the proposed framework aims to enhance the detection accuracy and efficiency of DDoS attacks in IoT environments. The following subsections detail the specific components and configurations of the proposed hybrid model, providing a comprehensive overview of the CNN, LSTM, and MHAM techniques utilized in this approach. A CNN is a DL model commonly used for tasks such as image processing, object recognition, facial recognition, video analysis, natural language processing, and intrusion detection. The structure of a CNN consists of several layers that execute different operations such as convolution, activation, normalization, pooling, and fully connected layers [Halbouni et al., 22]. The LSTM is a DL technique that captures long-term dependencies, building upon the Recurrent Neural Network (RNN) architecture. It employs a cell structure with three essential gates: input, forget, and output gates, which regulate data processing within the cell [Yu et al, 19; Graves, 12]. In a BiLSTM network, there are two distinct LSTM layers: the forward and the backward layers. The forward layer processes the input sequence in a sequential manner, from the first sample to the last sample. Conversely, the backward layer processes the input sequence in the reverse order, from the last sample back to the first sample. This bidirectional processing allows the BiLSTM network to capture both past and future context information for each time step in the sequence, enhancing its ability to model temporal dependencies effectively [Imrana et al., 21; Sri Vidhya and Nagarajan, 24].

MHAM typically operates within an Encoder-Decoder architecture in a transformer. This transformer architecture consists of an encoder and a decoder. However, in this study, only the encoder module is utilized, specifically adapted for network intrusion detection tasks where decoding is unnecessary. The Encoder processes input data, transforming it into a more suitable representation [Liu and Wu, 23]. The MHAM significantly boosts the capabilities of neural networks by enabling them to simultaneously focus on various segments of the input sequence. Unlike a single attention mechanism, MHAM employs multiple attention heads that operate simultaneously. Each attention head autonomously learns to attend to distinct facets of the input, thereby capturing a wide array of patterns and relationships present in the data [Long et al., 24]. The MHAM architecture addresses the limitations of traditional RNNs, such as challenges with parallel computation. Unlike CNNs, the structure of MHAM maintains equal spatial distances between features, thereby reducing the number of operations required to compute their associations. By incorporating MHAM and residual networks, the model achieves a deep architecture, enabling it to focus on temporal and detailed features across various subspaces.

Figure 4 illustrates the proposed hybrid DL model for classification in this study, comprising CNN, BiLSTM, and MHAM components. The CNN efficiently captures spatial information from the input data, while the BiLSTM processes the sequential information, capturing long-range dependencies in both forward and backward

directions. Additionally, the MHAM further refines feature representations by leveraging multi-head self-attention mechanisms and residual connections. This integration allows the model to effectively capture both spatial and temporal features across various subspaces, leading to improved accuracy and robustness in intrusion detection tasks.

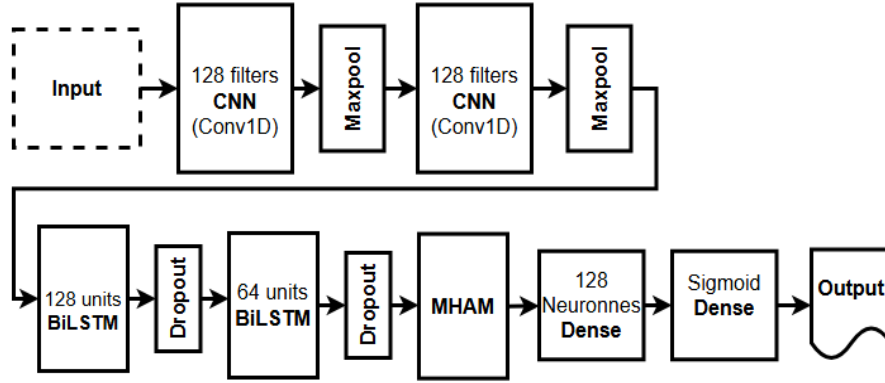


Figure 4: The proposed hybrid DL model for classification

As depicted in Figure 4, the proposed classification model presents a novel hybrid deep neural network architecture that integrates CNN, BiLSTM layers, and MHAM mechanisms. The hyperparameter optimization process employed a rigorous methodological approach utilizing the HyperOpt library with TPE algorithm, systematically exploring an extensive parameter space to identify optimal configurations for each architectural component.

The convolutional layers were optimized through a comprehensive evaluation of critical parameters. The search space encompassed filter quantities ranging from 64 to 256, kernel dimensions of 3×3 and 5×5 , and activation functions including ReLU, Exponential Linear Unit (ELU), and tanh. Architectural considerations included stride values of 1 or 2, padding schemes (valid versus same), and initialization methodologies incorporating both Glorot Uniform and He Normal approaches for kernel weights, with bias terms initialized to zero or unity values. The optimization process yielded an optimal configuration featuring 128 filters with 3×3 convolutional kernels, ReLU activation functions, unitary stride length, valid padding scheme, ‘He Normal’ kernel initialization, and zero-valued bias terms. Each convolutional layer incorporated batch normalization, with max pooling operations utilizing a 2×2 receptive field and valid padding configuration.

For the sequential processing components, the optimization framework evaluated BiLSTM architectures with hidden unit dimensions of 64, 128, and 256. Dropout regularization rates were sampled from a continuous uniform distribution spanning 0.2 to 0.5. The emergent optimal architecture comprised two BiLSTM layers with 128 and 64 hidden units respectively, implementing dropout rates of 0.43 and 0.39. The multi-head attention mechanism was configured through evaluation of model dimensionalities (64, 128, 256), attention head counts (2, 4, 8), and feed-forward

network sizes (64, 128, 256), with dropout rates uniformly distributed between 0.1 and 0.2. The selected configuration employed 64-dimensional models, 4 parallel attention heads, 256-unit feed-forward networks, and a dropout rate of 0.12.

The classification subsystem was optimized through evaluation of fully-connected layer dimensionalities (64, 128, 256) with dropout rates ranging from 0.2 to 0.5, culminating in the selection of a 128-unit layer with 0.4 dropout probability. The output layer employed a sigmoidal activation function (selected over softmax) with L1 and L2 regularization strengths sampled from the interval [0.001, 0.01], with optimal values of 0.004 and 0.009 respectively.

The training protocol optimization compared stochastic gradient descent approaches (Adam, SGD, RMSprop), loss functions (binary cross-entropy, mean squared error), and batch sizes (1024, 2048). The final configuration employed the Adam optimizer with binary cross-entropy objective function and a batch size of 2048 samples. This exhaustive hyperparameter optimization process, grounded in Bayesian optimization principles, has yielded a robust architectural configuration that optimally synthesizes spatial feature extraction, sequential pattern recognition, and attention mechanisms while maintaining computational efficiency through carefully calibrated regularization parameters. The resulting architecture demonstrates superior performance characteristics, achieving an optimal balance between model complexity and generalization capacity.

4 Experimental Analysis

The implementation was carried out on a Windows OS 64-bit system powered by an AMD Ryzen Pro 3400G processor, running at 3.70 GHz, and equipped with 16 GB of RAM. CBM-IDS was developed using a combination of several Python libraries, including Keras for building neural networks, TensorFlow for efficient computation serving as the backend for Keras, sklearn for ML algorithms and preprocessing, NumPy for numerical computations and array operations, Pandas for data manipulation and analysis, Matplotlib for data visualization, Fast-ML for feature selection and evaluation, and Hyperopt for hyperparameter optimization to enhance model performance. To evaluate the effectiveness of the CBM-IDS, various performance metrics, including the accuracy, precision, recall, and F1-score, along with evaluation tools such as the confusion matrix and receiver operating characteristics (ROC) curve. The evaluation was conducted on the CICDDoS2019 training dataset using stratified 5-fold CV. During training, the process incorporated Keras library callbacks such as “EarlyStopping” and “ReduceLROnPlateau” to optimize the model performance and efficiency. The “EarlyStopping” callback was configured with parameters `patience=20`, `mode='min'`, and `restore_best_weights=True`. This setting allows training to halt early if validation loss fails to improve for 20 consecutive epochs, thereby preventing overfitting and ensuring the model retains its best-performing weights. In addition, the “ReduceLROnPlateau” callback, configured with `monitor='val_loss'`, `factor=0.2`, `patience=10`, and `min_lr=0.001`, dynamically reduces the learning rate when validation loss plateaus. This adjustment helps the model converge more effectively by escaping local minima and improving overall performance. Five experiments are conducted to evaluate the performance of the CBM-IDS under these conditions.

Dimensionality reduction for CBM-IDS was performed using an autoencoder-ANOVA F-Test hybrid model, while data balancing was achieved with an ADASYN-OSS hybrid model. Classification was executed using a CNN-BiLSTM-MHAM hybrid model. Feature reduction initially involved the use of an autoencoder, followed by feature selection with the ANOVA F-Test. The resulting features were then used to train CBM-IDS separately with 10, 15, and 20 features. For cases with 25 features, feature reduction was performed exclusively using the autoencoder method, excluding the ANOVA F-Test. Additionally, CBM-IDS was trained using all 64 original features without dimensionality reduction. The proposed CNN-BiLSTM-MHAM hybrid method was employed for classification, and the accuracy results obtained from training with stratified 5-fold cross-validation, utilizing EarlyStopping and ReduceLROnPlateau callbacks, are presented in Table 2.

Features (F)	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean Accuracy
10F	0.99819	0.99829	0.99833	0.99841	0.99810	0.99826
15F	0.99941	0.99942	0.99936	0.99933	0.99945	0.99939
20F	0.99835	0.99813	0.99836	0.99854	0.99822	0.99832
25F	0.99809	0.99809	0.99858	0.99828	0.99800	0.99821
Original 64F	0.99816	0.99819	0.99836	0.99800	0.99800	0.99814

Table 2: Accuracy comparison of feature sets via 5-fold CV on training data

As seen in Table 2, the results show that using 15 features achieved the highest mean accuracy of 0.99939, with relatively fewer epochs required for training compared to models with more or fewer features. This suggests that 15 features strike an optimal balance between dimensionality reduction and retaining the essential information for classification. Comparing the performance across different numbers of features reveals that increasing the number of features beyond 15 does not significantly improve the accuracy and can even slightly reduce it. Additionally, the original set of 64 features resulted in a lower mean accuracy (0.99814) than the models using a reduced set of features. The model trained with 25 features, where only the autoencoder method was used for feature reduction without the ANOVA F-Test, achieved a mean accuracy of 0.99821. While this shows the potential of autoencoder in feature reduction, the hybrid approach combining autoencoder with ANOVA F-Test for 15 features still outperformed the other configurations. This underscores the efficacy of the hybrid feature selection method in enhancing the model's performance. These findings highlight the importance of feature selection in improving the efficiency and performance of the proposed DL model for CBM-IDS. Applying early stopping and learning rate reduction techniques also contributed to more effective training by preventing overfitting and optimizing the learning process. Table 3 presents the performance metrics of CBM-IDS on the test dataset with 10, 15, 20, 25, and the original 64 features. The metrics include accuracy, precision, recall, and F1-score.

Class	Metric	10F	15F	20F	25F	Original 64F
Benign	Accuracy	0.99786	0.99928	0.99798	0.99776	0.99766
	Precision	0.99486	0.99837	0.99536	0.99435	0.99420

	Recall	0.99572	0.99847	0.99577	0.99582	0.99551
	F1-Score	0.99529	0.99842	0.99557	0.99508	0.99486
Attack	Accuracy	0.99786	0.99928	0.99798	0.99776	0.99766
	Precision	0.99874	0.99955	0.99875	0.99877	0.99868
	Recall	0.99848	0.99952	0.99863	0.99833	0.99829
	F1-Score	0.99861	0.99953	0.99869	0.99855	0.99848

Table 3: Performance metrics of CBM-IDS on the test dataset

As seen in Table 3, the performance metrics of the proposed CNN-BiLSTM-MHAM hybrid classification model on the test dataset vary across different feature sets. Notably, the highest performance metrics are achieved with 15 features, indicating the optimal performance of the model in this configuration. While all feature sets yield strong results, a one-way ANOVA test on F1-scores indicated that the improvement achieved with the 15-feature configuration is statistically more significant compared to the other subsets. Additionally, Figure 5 illustrates the confusion matrix used to evaluate the performance of CBM-IDS on the test dataset with these 15 features.

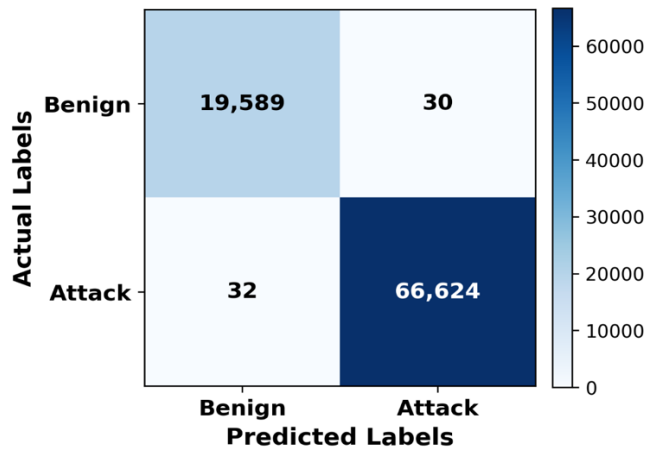


Figure 5: Confusion matrix for CBM-IDS on test dataset with 15 features

The confusion matrix in Figure 5 presents the counts of true negatives (TN), false positives (FP), false negatives (FN), and true positives (TP), offering valuable insights into the classification performance of the model. In this case, it reveals that there are 19589 true negatives, indicating correctly predicted benign instances; 30 false negatives, which are benign instances incorrectly predicted as attacks; 32 false positives, representing attack instances incorrectly predicted as benign; and 66624 true positives, denoting correctly predicted attack instances.

During the inference phase, the average prediction time per network traffic instance was measured to be under 10 milliseconds. These results demonstrate that the proposed CBM-IDS architecture achieves a level of computational efficiency sufficient for real-time intrusion detection applications. Therefore, the model exhibits strong potential for deployment both in centralized settings and especially in distributed architectures

commonly used in IoT environments such as edge and fog computing infrastructures where low latency and high throughput are critical.

To assess the effectiveness of the proposed Autoencoder–ANOVA F-test feature reduction method, it was compared with Recursive Feature Elimination (RFE) and Mutual Information (MI) techniques using the CBM-IDS model. The classification results are illustrated in Table 4.

Method	Accuracy	Precision	Recall	F1-Score
Autoencoder+ANOVA F-Test	0.99928	0.99896	0.99899	0.99897
RFE	0.99816	0.99689	0.99787	0.99738
MI	0.99553	0.99282	0.99448	0.99364

Table 4: Performance comparison of feature selection methods

As shown in Table 4, the proposed method achieved the highest accuracy (99.928%) and F1-score (99.897%), outperforming both RFE and MI. These results demonstrate its superior ability to retain relevant features and enhance classification performance.

To evaluate the effectiveness of data balancing strategies on the classification performance of the proposed CBM-IDS model, a comparative analysis was conducted. In real-world IoT network traffic, data imbalance between benign and attack instances is a common challenge that can negatively affect the learning capability and generalization performance of IDSs. Figure 6 illustrates the comparison between the classification performance of CBM-IDS using these 15 features, with and without data balancing.

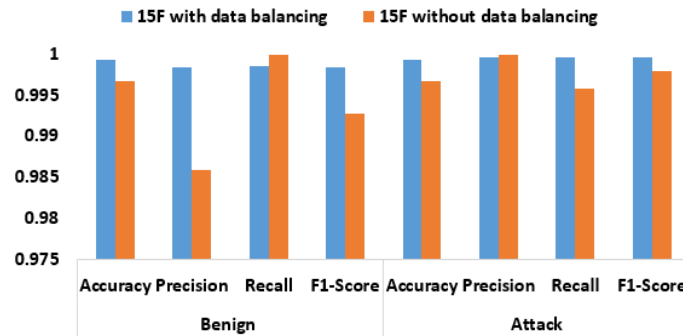


Figure 6: Performance comparison of models with and without data balancing

The comparison in Figure 6 shows the significant influence of the ADASYN-OSS data balancing technique on the performance of CBM-IDS. When comparing scenarios with and without data balancing across 15 features, where the highest performance was achieved, it becomes evident that data balancing has a considerable impact on performance metrics. With data balancing, the accuracy of both the benign and attack classes improve by approximately 0.257 %. Precision increases by approximately 1.449% for the benign class and 1.460% for the attack class, underscoring the enhanced ability to correctly identify true positives. Recall, however, shows a slight decrease of

approximately -0.138% for the benign class but an increase of 0.374% for the attack class, indicating better detection of actual positive instances in the latter. The F1-score, which balances precision and recall, improves by 0.561% for the benign class and 0.166% for the attack class with data balancing. These numerical differences highlight the crucial role of data imbalance mitigation techniques in enhancing the performance of CBM-IDS. While there is a minor trade-off in recall for the benign class, the overall gains in accuracy, precision, and F1-score demonstrate the positive impact of data balancing on the effectiveness of CBM-IDS in identifying and responding to intrusions.

The generalization capability of the proposed CBM-IDS model was evaluated on the IoTID20 dataset. A two-stage feature reduction approach was employed, where an Autoencoder first compressed the original 83 features to 42, followed by ANOVA F-Test selecting the 15 most relevant features. Class imbalance in the training set was addressed using the ADASYN-OSS hybrid technique. Classification was then performed using the CNN-BiLSTM-Transformer model. The resulting confusion matrix is depicted in Figure 7.

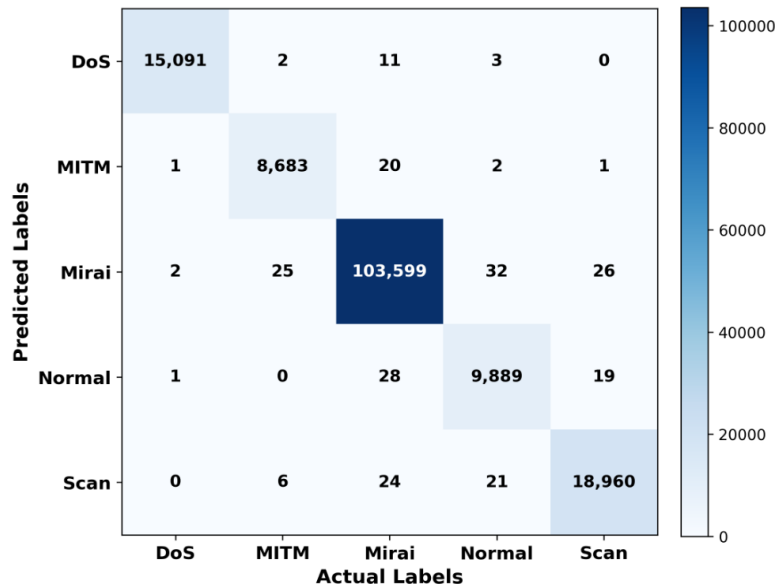


Figure 7: Confusion matrix for CBM-IDS on IoTID20 dataset

Based on Figure 7, which shows the confusion matrix, the model achieved an accuracy of 0.99982, precision of 0.99921, recall of 0.99899, and an F1-score of 0.99905. These results confirm CBM-IDS's robustness and effectiveness across different IoT datasets.

In the study, the performance of various traditional ML algorithms, including Random Forest (RF), Logistic Regression (LR), Decision Tree (DT), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Gaussian Naive Bayes (Gaussian NB), was also evaluated on CICDDoS2019 test dataset comprising 15 features obtained after feature reduction and data balancing. Additionally, the default hyperparameter values provided by the sklearn library were utilized for these ML

methods. RF utilized 100 estimators with a “gini” criterion and no maximum depth limit, while LR employed an “l2” penalty, a tolerance value of 0.0001, and a regularization strength (C) of 1.0. DT utilized the “gini” criterion and “best” splitter, whereas KNN employed 5 neighbors with uniform weighting and the “auto” algorithm. SVM utilized a regularization parameter (C) of 1.0, an “rbf” kernel, and a polynomial kernel degree of 3. NB employed a smoothing parameter called “var_smoothing” with a default setting of 1e-9. The results are shown in Table 5.

Model	Accuracy	Precision	Recall	F1-Score
RF	0.99891	0.99951	0.99906	0.99929
LR	0.98919	0.99209	0.99393	0.99301
DT	0.99835	0.99893	0.99893	0.99893
KNN	0.99862	0.99929	0.99891	0.99910
SVM	0.99638	0.99831	0.99699	0.99765
NB	0.94088	0.97280	0.95004	0.96129

Table 5: Performance metrics for traditional ML models

As seen in Table 5, traditional ML models such as RF, DT, and KNN attained high accuracy scores ranging from approximately 99.83% to 99.89% on the CICDDoS2019 dataset. In contrast, the proposed hybrid DL model, which combines CNN, BiLSTM, and MHAM architectures, outperformed traditional ML models. It achieved the highest accuracy, precision, recall, and F1-score for both the benign and attack classes, making it the most effective approach for intrusion detection.

In this study, the performance evaluation results demonstrate the detection capabilities of the proposed hybrid CNN-BiLSTM-Transformer model compared to its individual components, as well as combinations of these components taken in pairs, as shown in Table 6.

Model	Accuracy	Precision	Recall	F1-Score
CNN-BiLSTM-Transformer	0.99928	0.99896	0.99899	0.99897
BiLSTM-Transformer	0.99843	0.99745	0.99546	0.99334
CNN-Transformer	0.99815	0.99758	0.99562	0.99468
CNN-BiLSTM	0.99833	0.99759	0.99597	0.99471

Table 6: Performance metrics for DL models

As seen in Table 6, the complete hybrid architecture outperforming all other configurations. Additionally, in the study, the performance of CBM-IDS in terms of accuracy was compared to recent studies in the literature using the CICDDoS2019 dataset. The results of this comparison are presented in Table 7.

Study	Method	Accuracy
[Almiani et al., 21]	Kalman backpropagation neural network	94.00%
[Alghazzawi et al., 21]	CNN, BiLSTM	94.52%
[Roopak et al., 24]	K-means-GMM, One class SVM	94.55%

[Rajagopal et al., 21]	Extended DT	97.00%
[Wei et al., 21]	AE, MLP	98.34%
[Nie et al., 21]	Generative adversarial network (GAN)	98.53%
[Wu et al., 22]	Transformer	98.58%
[Hussan et al., 23]	Elman recurrent neural network (ERNN)	98.64%
[Benmohamed et al. 24]	Stacked/bagged MLP	98.86%
[Kumar et al., 24]	Deep Residual CNN (DRCNN)	99.12%
[Aslam et al., 23]	XGBoost	99.50%
[Zainudin et al., 22]	CNN, LSTM	99.50%
[Li et al., 23]	Tri-broad learning system (TBLS)	99.51%
[Maiga et al., 23]	LSTM, BiGRU, BiLSTM	99.59%
CBM-IDS	CNN, BiLSTM, MHAM	99.93%

Table 7: Accuracy comparisons of CBM-IDS with earlier research

As seen in Table 7, CBM-IDS in this study outperforms all other methods in the literature with an accuracy of 99.93%. This result demonstrates the superior capability of CBM-IDS for detecting and classifying attacks. Other methods in the literature show accuracy rates ranging from 94% to 99.59%, indicating that CBM-IDS provides a significant performance improvement and stands out as the most effective solution for intrusion detection.

5 Conclusions and Future Work

The rise of IoT devices has brought about numerous advantages, but it has also exposed them to considerable security vulnerabilities. These devices typically have limited computational capabilities and memory resources, which present challenges in implementing robust security measures. Furthermore, the diverse communication protocols and the vast amount of data they handle make them attractive targets for cyber-attacks. DDoS attacks are among the most critical threats faced by the IoT devices. These attacks flood the devices with excessive traffic, overwhelming their resources and rendering them inoperative. The impact of DDoS attacks on IoT networks can be devastating, leading to service disruptions and potentially compromising the sensitive data.

In this study, the CICDDoS2019 dataset, which includes contemporary DDoS attack behaviors, was used to evaluate CBM-IDS. The experiments conducted in this study aimed to evaluate the effectiveness of a hybrid DL model for intrusion detection, along with the impact of feature reduction and data balancing techniques on model performance. To reduce the dimensionality of the dataset, a hybrid model combining autoencoder and the ANOVA F-test was employed. A hybrid model using ADASYN and OSS was introduced to balance the dataset. For classification, a hybrid DL model integrating CNN, BiLSTM, and MHAM architectures was proposed. The results demonstrated that the proposed hybrid model significantly outperformed state-of-the-art ML methods and existing approaches in the literature. Specifically, after implementing dimensionality reduction and data balancing techniques, classification using the CBM-IDS hybrid model achieved an accuracy of 99.93%, surpassing the

accuracy rates ranging from 94% to 99.59% observed in previous studies. The advanced architecture of the model enables it to capture complex patterns and behaviors associated with modern DDoS attacks, making it a more robust solution for improving the security of IoT devices.

In future works, it is planned to evaluate CBM-IDS in real-world IoT environments to assess its practical applicability. The scope of the study will also be extended to explore the potential of transfer learning and the feasibility of deploying the model on edge devices, which is particularly critical for IoT applications with limited computational resources. Furthermore, future research will aim to evaluate the system's resilience against adversarial attacks and concept drift.

References

- [Ahmad et al., 21] Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., and Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, 32(1), 1-29.
- [Ahmim et al., 23] Ahmim, A., Maazouzi, F., Ahmim, M., Namane, S., and Dhaou, I. B. (2023). Distributed denial of service attack detection for the Internet of Things using hybrid deep learning model. *IEEE Access*, 11, 119862–119875.
- [Alghazzawi et al., 21] Alghazzawi, D., Bamasag, O., Ullah, H., and Asghar, M. Z. (2021). Efficient detection of DDoS attacks using a hybrid deep learning model with improved feature selection. *Applied Sciences*, 11(24), 11634.
- [Almiani et al., 21] Almiani, M., AbuGhazleh, A., Jararweh, Y., and Razaque, A. (2021). DDoS detection in 5G-enabled IoT networks using deep Kalman backpropagation neural network. *International Journal of Machine Learning and Cybernetics*, 12(11), 3337–3349.
- [Alotaibi and Ilyas, 23] Alotaibi, Y., and Ilyas, M. (2023). Ensemble-learning framework for intrusion detection to enhance internet of things' devices security. *Sensors*, 23(12), 5568.
- [Alwahedi et al., 24] Alwahedi, F., Aldhaheer, A., Ferrag, M. A., Battah, A., and Tihanyi, N. (2024). Machine learning techniques for IoT security: Current research and future vision with generative AI and large language models. *Internet of Things and Cyber-Physical Systems*.
- [Aslam et al., 23] Aslam, N., Srivastava, S., and Gore, M. M. (2023). ONOS DDoS Defender: A Comparative Analysis of Existing DDoS Attack Datasets using Ensemble Approach. *Wireless Personal Communications*, 133(3), 1805–1827.
- [Bacha et al., 24] Bacha, S., Aljuhani, A., Abdellafou, K. B., Taouali, O., Liouane, N., and Alazab, M. (2024). Anomaly-based intrusion detection system in IoT using kernel extreme learning machine. *Journal of Ambient Intelligence and Humanized Computing*, 15(1), 231–242.
- [Batchu and Seetha, 22] Batchu, R. K., and Seetha, H. (2022). On improving the performance of DDoS attack detection system. *Microprocessors and Microsystems*, 93, 104571.
- [Benmohamed et al., 24] Benmohamed, E., Thaljaoui, A., Elkhediri, S., Aladhadh, S., and Alohal, M. (2024). E-SDNN: encoder-stacked deep neural networks for DDOS attack detection. *Neural Computing and Applications*, 36, 10431–10443.
- [Bergstra et al., 13] Bergstra, J., Yamins, D., and Cox, D. D. (2013). Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. *Proc. of the 30th International Conference on Machine Learning (ICML)*, 28(1), 115–123.

- [Bommert et al., 20] Bommert, A., Sun, X., Bischl, B., Rahnenführer, J., and Lang, M. (2020). Benchmark for filter methods for feature selection in high-dimensional classification data. *Computational Statistics and Data Analysis*, 143, 106839.
- [Cao et al., 22] Cao, B., Li, C., Song, Y., Qin, Y., and Chen, C. (2022). Network Intrusion Detection Model Based on CNN and GRU. *Applied Sciences*, 12(9), 4184.
- [Cao et al., 23] Cao, Y., Wang, Z., Ding, H., Zhang, J., and Li, B. (2023). An intrusion detection system based on stacked ensemble learning for IoT network. *Computers and Electrical Engineering*, 110, 108836.
- [Chaabouni et al., 19] Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., and Faruki, P. (2019). Network intrusion detection for IoT security based on learning techniques. *IEEE Communications Surveys and Tutorials*, 21(3), 2671–2701.
- [Chawla et al., 02] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357.
- [de Souza et al., 20] de Souza, C. A., Westphall, C. B., Machado, R. B., Sobral, J. B. M., and dos Santos Vieira, G. (2020). Hybrid approach to intrusion detection in fog-based IoT environments. *Computer Networks*, 180, 107417.
- [Du et al., 23] Du, J., Yang, K., Hu, Y., and Jiang, L. (2023). NIDS-CNNLSTM: Network intrusion detection classification model based on deep learning. *IEEE Access*, 11, 24808–24821.
- [Gökdemir and Calhan, 22] Gökdemir, A., and Calhan, A. (2022). Deep learning and machine learning based anomaly detection in internet of things environments. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 37(4), 1945–1956.
- [Halbouni et al., 22] Halbouni, A., Gunawan, T. S., Habaebi, M. H., Halbouni, M., Kartiwi, M., and Ahmad, R. (2022). CNN-LSTM: hybrid deep neural network for network intrusion detection system. *IEEE Access*, 10, 99837–99849.
- [He et al., 08] He, H., Bai, Y., Garcia, E. A., and Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 1322–1328.
- [Hussan et al., 23] Hussan, M. T., Reddy, G. V., Anitha, P. T., Kanagaraj, A., and Naresh, P. (2023). DDoS attack detection in IoT environment using optimized Elman recurrent neural networks based on chaotic bacterial colony optimization. *Cluster Computing*, 1–22.
- [Imrana et al., 21] Imrana, Y., Xiang, Y., Ali, L., and Abdul-Rauf, Z. (2021). A bidirectional LSTM deep learning approach for intrusion detection. *Expert Systems with Applications*, 185, 115524.
- [Keserwani et al., 21] Keserwani, P. K., Govil, M. C., Pilli, E. S., and Govil, P. (2021). A smart anomaly-based intrusion detection system for the Internet of Things (IoT) network using GWO–PSO–RF model. *Journal of Reliable Intelligent Environments*, 7(1), 3–21.
- [Krishnaveni et al., 21] Krishnaveni, S., Sivamohan, S., Sridhar, S. S., and Prabakaran, S. (2021). Efficient feature selection and classification through ensemble method for network intrusion detection on cloud computing. *Cluster Computing*, 24(3), 1761–1779.
- [Kubat and Matwin, 97] Kubat, M., and Matwin, S. (1997). Addressing the course of imbalanced training sets: One-sided selection. *ICML*, 179–186.

- [Kumar et al., 24] Kumar, G. S. C., Kumar, R. K., Kumar, K. P. V., Sai, N. R., and Brahmaiah, M. (2024). Deep residual convolutional neural Network: An efficient technique for intrusion detection system. *Expert Systems with Applications*, 238, 121912.
- [Kunang et al., 21] Kunang, Y. N., Nurmaini, S., Stiawan, D., and Suprpto, B. Y. (2021). Attack classification of an intrusion detection system using deep learning and hyperparameter optimization. *Journal of Information Security and Applications*, 58, 102804.
- [Labiod et al., 22] Labiod, Y., Amara Korba, A., and Ghoulmi, N. (2022). Fog Computing-Based Intrusion Detection Architecture to Protect IoT Networks. *Wireless Personal Communications*, 125, 231–259.
- [Laghrissi et al., 21] Laghrissi, F., Douzi, S., Douzi, K., and Hssina, B. (2021). Intrusion detection systems using long short-term memory (LSTM). *Journal of Big Data*, 8(1), 65.
- [Li et al., 23] Li, J., Zhang, H., Liu, Z., and Liu, Y. (2023). Network intrusion detection via tri-broad learning system based on spatial-temporal granularity. *The Journal of Supercomputing*, 79(8), 9180–9205.
- [Liu et al., 21] Liu, J., Gao, Y., and Hu, F. (2021). A fast network intrusion detection system using adaptive synthetic oversampling and LightGBM. *Computers and Security*, 106, 102289.
- [Liu and Wu, 23] Liu, Y., and Wu, L. (2023). Intrusion Detection Model Based on Improved Transformer. *Applied Sciences*, 13(10), 6251.
- [Long et al., 24] Long, Z., Yan, H., Shen, G., Zhang, X., He, H., and Cheng, L. (2024). A Transformer-based network intrusion detection approach for cloud security. *Journal of Cloud Computing*, 13(1), 5.
- [Mandrekar, 10] Mandrekar, J. N. (2010). Receiver operating characteristic curve in diagnostic test assessment. *J Thoracic Oncol*, 5(9), 1315–1316.
- [Martins et al., 22] Martins, I., Resende, J. S., Sousa, P. R., Silva, S., Antunes, L., and Gama, J. (2022). Host-based IDS: A review and open issues of an anomaly detection system in IoT. *Future Generation Computer Systems*, 133, 95–113.
- [Miaga et al., 23] Miaga, A., Ataro, E., and Githinji, S. (2023). Secured federated learning for DDoS detection in heterogeneous telecom cloud networks using recurrent neural networks. *SSRG International Journal of Electrical and Electronics Engineering*, 10(12), 54–64.
- [Mighan and Kahani, 21] Mighan, S. N., and Kahani, M. (2021). A novel scalable intrusion detection system based on deep learning. *International Journal of Information Security*, 20(3), 387–403.
- [Mohajer et al., 13a] Mohajer, A., Bavaghar, M., Saboor, R., and Payandeh, A. (2013). Secure dominating set-based routing protocol in MANET: Using reputation. *10th International ISC Conference on Information Security and Cryptology (ISCISC)*, 1–7.
- [Mohajer et al., 13b] Mohajer, A., Mazoochi, M., Niasar, F. A., Ghadikolayi, A. A., and Nabipour, M. (2013). *20th International Conference on Computer Networks (CN 2013)*, 277–289.
- [Mohajer et al., 24] Mohajer, A., Hajipour, J., and Leung, V. C. (2024). Dynamic offloading in mobile edge computing with traffic-aware network slicing and adaptive TD3 strategy. *IEEE Communications Letters*, 29(1), 95–99.
- [Najafimehr et al., 22] Najafimehr, M., Zarifzadeh, S., and Mostafavi, S. (2022). A hybrid machine learning approach for detecting unprecedented DDoS attacks. *The Journal of Supercomputing*, 78(7), 8106–8136.

- [Nie et al., 21] Nie, L., Wu, Y., Wang, X., Guo, L., Wang, G., Gao, X., and Li, S. (2021). Intrusion detection for secure social internet of things based on collaborative edge computing: A generative adversarial network-based approach. *IEEE Transactions on Computational Social Systems*, 9(1), 134–145.
- [Otoum and Nayak, 21] Otoum, Y., and Nayak, A. (2021). AS-IDS: Anomaly and signature based IDS for the Internet of Things. *Journal of Network and Systems Management*, 29(23).
- [Ozkan-Ozay et al., 24] Ozkan-Ozay, M., Akin, E., Aslan, Ö., Kosunalp, S., Iliev, T., Stoyanov, I., and Beloev, I. (2024). A Comprehensive Survey: Evaluating the Efficiency of Artificial Intelligence and Machine Learning Techniques on Cyber Security Solutions. *IEEE Access*.
- [Rajagopal et al., 21] Rajagopal, S., Kundapur, P. P., and Hareesha, K. S. (2021). Towards effective network intrusion detection: From concept to creation on Azure cloud. *IEEE Access*, 9, 19723–19742.
- [Roopak et al., 23] Roopak, M., Parkinson, S., Tian, G. Y., Ran, Y., Khan, S., and Chandrasekaran, B. (2023). An Unsupervised Approach for the Detection of Zero-Day DDoS Attacks in IoT Networks (pp. 1–9).
- [Saba et al., 22] Saba, T., Rehman, A., Sadad, T., Kolivand, H., and Bahaj, S. A. (2022). Anomaly-based intrusion detection system for IoT networks through deep learning model. *Computers and Electrical Engineering*, 99, 107810.
- [Saiyed and Al-Anbagi, 24] Saiyed, M. F., and Al-Anbagi, I. (2024). Deep Ensemble Learning with Pruning for DDoS Attack Detection in IoT Networks. *IEEE Transactions on Machine Learning in Communications and Networking*, 2, 596–616.
- [Salim et al., 20] Salim, M. M., Rathore, S., and Park, J. H. (2020). Distributed denial of service attacks and its defenses in IoT: a survey. *The Journal of Supercomputing*, 76, 5320–5363.
- [Samha et al., 23] Samha, A. K., Malik, N., Sharma, D., and Dutta, P. (2023). Intrusion detection system using hybrid convolutional neural network. *Mobile Networks and Applications*, 1–13.
- [Sayegh et al., 24] Sayegh, H. R., Dong, W., and Al-madani, A. M. (2024). Enhanced Intrusion Detection with LSTM-Based Model, Feature Selection, and SMOTE for Imbalanced Data. *Applied Sciences*, 14(2), 479.
- [Sharafaldin et al., 19] Sharafaldin, I., Lashkari, A. H., Hakak, S., and Ghorbani, A. A. (2019). Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy. 2019 International Carnahan Conference on Security Technology (ICCST), 1–8.
- [Sri Vidhya and Nagarajan, 24] Sri Vidhya, G., and Nagarajan, R. (2024). A novel bidirectional LSTM model for network intrusion detection in SDN-IoT network. *Computing*, 1–30.
- [Talukder and Uddin, 23] Talukder, M. A., and Uddin, M. A. (2023). CIC-DDoS2019 Dataset.
- [Thapa et al., 20] Thapa, N., Liu, Z., Kc, D. B., Gokaraju, B., and Roy, K. (2020). Comparison of machine learning and deep learning models for network intrusion detection systems. *Future Internet*, 12(10), 167.
- [Varoquaux et al., 15] Varoquaux, G., Buitinck, L., Louppe, G., Grisel, O., Pedregosa, F., and Mueller, A. (2015). Scikit-learn: Machine learning without learning the machinery. *GetMobile: Mobile Computing and Communications*, 19(1), 29–33.
- [Vibhute et al., 24] Vibhute, A. D., Khan, M., Kanade, A., Patil, C. H., Gaikwad, S. V., Patel, K. K., and Saini, J. R. (2024). An LSTM-based novel near-real-time multiclass network intrusion detection system for complex cloud environments. *Concurrency and Computation: Practice and Experience*, 36(11).

- [Wang et al., 22] Wang, C., Liu, H., Sun, Y., Wei, Y., Wang, K., and Wang, B. (2022). Dimension reduction technique based on supervised autoencoder for intrusion detection of industrial control systems. *Security and Communication Networks*, 2022(1), 5713074.
- [Wei et al., 21] Wei, Y., Jang-Jaccard, J., Sabrina, F., Singh, A., Xu, W., and Camtepe, S. (2021). Ae-mlp: A hybrid deep learning approach for DDoS detection and classification. *IEEE Access*, 9, 146810–146821.
- [Wu et al., 22] Wu, Z., Zhang, H., Wang, P., and Sun, Z. (2022). RTIDS: A robust transformer-based approach for intrusion detection system. *IEEE Access*, 10, 64375–64387.
- [Xiao et al., 19] Xiao, Y., Xing, C., Zhang, T., and Zhao, Z. (2019). An intrusion detection model based on feature reduction and convolutional neural networks. *IEEE Access*, 7, 42210–42219.
- [Yu et al., 19] Yu, Y., Si, X., Hu, C., and Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. *Neural Computation*, 31(7), 1235–1270.
- [Zainudin et al., 22] Zainudin, A., Ahakonye, L. A. C., Akter, R., Kim, D. S., and Lee, J. M. (2022). An efficient hybrid-DNN for DDoS detection and classification in software-defined IIoT networks. *IEEE Internet of Things Journal*, 10(10), 8491–8504.
- [Zheng et al., 20] Zheng, D., Hong, Z., Wang, N., and Chen, P. (2020). An improved LDA-based ELM classification for intrusion detection algorithm in IoT application. *Sensors*, 20(6), 1706.
- [Zhu and Liu, 24] Zhu, J., and Liu, X. (2024). An integrated intrusion detection framework based on subspace clustering and ensemble learning. *Computers and Electrical Engineering*, 115, 109113.