

VISUAL FEATURE MAPPING AND FEATURE SELECTION FOR ROBUST IDS BASED ON CSE-CIC-IDS2018 DATASET

Satishkumar D. Prajapati^{*1}, Priyesh P. Gandhi², Sheshang Degadwala³,
Keyur Brahmabhatt⁴, Swapnil M Parikh⁵

¹Computer Engineering, Gujarat Technological University, Ahmedabad, Gujarat, India.

²Provost, Sigma University, Vadodara, Gujarat, India.

³Department of Computer Engineering, Sigma University, Vadodara, Gujarat, India.

⁴Department of Information and Technology, Birla Vishvakarma mahavidyalaya, Vallabh Vidhyanagar, Gujarat, India.

⁵Computer Science and Engineering, Parul Institute of Technology, Parul University, Vadodara, Gujarat, India.

¹<https://orcid.org/0009-0008-1349-3082>, ²<https://orcid.org/0000-0001-5338-3111>, ³<https://orcid.org/0000-0002-2385-7790>,

⁴<https://orcid.org/0000-0001-7478-3173>, ⁵<https://orcid.org/0009-0005-8816-5508>

E-mail: prajapati.satish90@gmail.com, priyeshgandhi@gmail.com, sheshang13@gmail.com, keyur.brahmbhatt@bvmengineering.ac.in

ARTICLE INFO

Article History

Received: March 13, 2026

Reviewed: April 16, 2026

Accepted: July 16, 2026

Published: August 31, 2026

Keywords:

Distributed Denial of Service,
Internet of Things,
Recursive Feature Elimination,
Ensemble Machine Learning,
Intrusion Detection Systems.

ABSTRACT

The rapid growth of Internet of Things (IoT) ecosystems posed a great risk to ubiquitous access to advanced cyber threats and thus the design of effective Intrusion Detection Systems (IDS) has become critical in ensuring the safeguard of secure network structures. Nonetheless, most classic deep learning methods, especially Recurrent Neural Networks (RNNs) tend to exhibit low generalization and low interpretability to tabular network traffic data and may fail to discriminate between different and intricate attacks. To overcome this issue, this paper presents a visual feature-based IDS model, which converts network traffic characteristics in the form of tabular features into RGB matrix images to harness the power of effective computer vision algorithms to detect intrusions. Recursive Feature Elimination (RFE) is applied to determine the best informative feature before generating an image, enhance features and decrease redundancy. The acquired visual representations are subsequently categorized by the EfficientNetB0 design by way of transfer learning. Experimental testing shows that the suggested method has a 98 percent accuracy, precision, recall, and F1-score, Matthews Correlation Coefficient (MCC) of 0.97, and AUC-ROC of 0.99. Comparatively, the RNN original does not reach the same level as it only gives 89% on performance metrics with MCC of 0.88 and AUC-ROC of 0.88. The above findings indicate the ability of feature selection combined with visual deep learning to facilitate scalable, precise, and real-time intrusion detection of modern-day IoT environments.



Copyright ©2026 by authors and Galileo Institute of Technology and Education of the Amazon (ITEGAM). This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

I. INTRODUCTION

Through the accelerated development of network systems and the spread of the Internet of Things (IoT) gadgets, the area of potential cyberattacks has increased dramatically. With the growing number of businesses and people depending on interconnected infrastructures, it has become part of the priorities to secure these systems against different threats to include Distributed Denial of Service (DDoS), probing, user-to-root (U2R), and remote-to-local (R2L) attacks. IDSS is an important tool that is used to observe the network traffic and detect dangerous actions as a line of defense after firewalls and antivirus programs. Nevertheless, the increasing complexity of cyberattacks is posing significant threats to more traditional methods of IDS, particularly those that are based on fixed rules or other more traditional machine learning classifiers that tend to be weak in terms of scalability, adaptability and high dimensional features. New developments in the field of deep learning created new opportunities to enhance the capabilities of IDS, especially the application of the Convolutional Neural Networks (CNNs) and the Transfer Learning models.

Such techniques have proven to be highly successful in many fields, particularly computer vision problems because they have strong powers of representation learning. Regarding IDS, there is a growing interest in using such methods in order to transform structured tabular network data into image formats that CNNs can process efficiently. This data-to-image encoding conversion provides the ability to encode latent geometric and contextual correlations of the traffic features that may be missed in the data with an older algorithm. The approach is even more powerful when paired with pre-trained models such as EfficientNet, since it embraces both hierarchies of features and efficiency. However, it is not so easy to transform raw network traffic into meaningful image representations. Among the main problems is the large dimensionality and sparsity of IDS data, including CSE-CIC-IDS2018, which have many redundant or irrelevant features. Such extraneous properties have the adverse effect of affecting the performance of deep learning models and lead to overfitting, high computational cost, and poor detection accuracy. Thus, before transformation is required, there must be effective feature selection. Recursive Feature Elimination (RFE) has been one of the most used and interpretable feature selection algorithms where less important features are dropped out one at a time due to model weight coefficients. With RFE, not all dimensions of the dataset are required to be retained, the most relevant ones are kept so as to visual mapping is done. This study suggests a new hybrid IDS model that integrates recursive feature selection, visual feature encoding as well as transfer learning. Preprocessing steps include the removal of null values, label encoding and normalization which is the beginning of the pipeline.

The smaller feature space is then coded into 64x64RGB images where temporal and structural relationships can be maintained as pixel intensity values. Through these images, a pre-trained EfficientNetB0 model is fine-tuned and used to execute binary and multi-class intrusion detection using the scalable nature of its architecture. The proposed model is measured against the standard measures of accuracy, precision, recall, F1-score, Matthews Correlation Coefficient (MCC), and Area Under the Receiver Operating Characteristic Curve (AUC-ROC). The main findings of this research are triple: (1) the effective data-to-image transformation of network traffic based on the RFE-selected features, (2) the application of EfficientNetB0 to deep visual feature extraction in the area of the IDS, and (3) the comparison of the model with the traditional structures (RNN) to prove the best results. After intensive testing of the presented system on the CSE-CIC-IDS2018 dataset, the results are encouraging, and the proposed system has accuracy and F1-score, which are over 98, much higher than the results of the baseline models. The rest of this paper will be organized as follows: Section 2 shows the related literature in the area of intrusion detection with the help of machine-learning tools and image-based techniques. Section 3 outlines the proposed methodology that comprises data preprocessing, feature selection, image conversion and model training. The 4th section will offer the performance analysis and results of the experiment. Lastly, Section 5 wraps up the research and gives the future directions of the research.

II. THEORETICAL REFERENCE

The recent past has seen a boom in the research on optimizing the efficiency, robustness, and accuracy of intrusion detection systems (IDS), especially by incorporating deep learning and hybrid frameworks. This part of the paper is a review of notable works in this field and the research gaps that have been identified that lead to the application of the proposed work. Markkandeyan et al. suggested a new hybrid deep learning model coupled with an optimization algorithm in [1] to enhance the ability to detect cyber threats. Their model showed improvement in detection performance, and no investigation was carried out on visual encoding of features or transfer learning adaptability. In the same manner, [2] have dealt with the issue of DDoS detection based on adaptive transfer learning. Although they are useful in dynamic network setting, their analysis failed to investigate recursive feature selection or feature visual representation, which might result in an even better generalization. By [3] compared several bio-inspired optimization algorithms on the CSE-CIC-IDS2018 dataset. Their research was useful in comparison but focused only on the traditional numerical features and did not rely on the deep visual feature extraction. In turn [4] were concerned with the machine learning-based IDSs adapted to cloud networks. Despite its practicality, the study did not have a dimension reduction procedure like the recursive feature selection to deal with the redundant features. According to [5] focused on cross-dataset generalization in IDS by performing cross-dataset performance analysis. Although their results have emphasized generalization, they have not examined how transfer learning with image-based inputs can influence robustness. By [6] applied deep learning to Software Defined Networks (SDNs) to identify intrusion, and the flexible character of the DL technique in the existing network architecture but did not consider the improvement of performance through CNN-based visual encoding. In turn [7] proposed a transformer-based IDS with transfer learning to solve the imbalance in traffic data. Though this model had good performance, the absence of a systematic elimination of features and lack of a hybrid pipeline that combined the use of CNNs did not make the model scalable.

According to [8], in their turn, performed a general review of IDS datasets, where the data quality and structure variation also prevalent, yet they did not carry out their analysis until the usefulness of visual transformations to standardized processing. By [9] have introduced a working model of the IDS that works in cloud computing, which is an integration of detection accuracy and low latency. Nevertheless, they failed to combine the visual aspect and recursive selection to determine the best feature subset. According to [10] suggested the use of DTL-IDS, which utilizes the deep transfer learning and genetic algorithm. Although the study showed good outcomes, it was not focused on image representations or effective feature filtering methods. In turn [11] addressed the issue of class imbalance in the cloud setting with deep learning. However, visual feature maps or multi-modal representation integration were not studied. By [12] offered an overall deep learning-based IDS framework that is well-accurate, though not scalable to further dimensions and complexities of features. In turn [13] presented a collaborative weighted ensemble DNN to coordinated attacks, which improved detection of heterogeneous networks. Nevertheless, their approach did not feature any techniques of transforming features to visual modalities. In turn [14] used contrastive variational autoencoders and metric learning to detect attacks in blockchain-enhanced cloud of things, demonstrating the power of deep generative models, but lacking the incorporation of classic feature selection techniques. According to [15] adopted a portable deep learning architecture to be applied to IDS in the IoT. Their approach, though promising, did not look into the possibility of combining recursive feature selection with visual encoding of CNNs. In turn [16] designed a new IoT IDS based on the deep learning algorithm against DDoS attacks. They concentrated on the type and model tuning of the attacks that they did and ignored the effect of the preprocessing pipeline.

To deal with the issue of data imbalance, [17] came up with a multi-classification model which incorporates reconstruction and feature matching. Though successful, they did not use recursive feature elimination or CNN-based visual learning in their model. By [18] created a deep learning IDS on the cloud platform, but the scheme did not have the hybrid feature extraction or transfer learning parts. According to [19] extensively examined the CIC-IDS2017 data but did not provide information on the CSE-CIC-IDS2018, as well as the consequences of feature changes. By [20] employed the optimized CNNs transfer learning on cloud-IoT IDS. Their design allows deep extraction of features though its feature selection mechanisms are absent which affects the efficiency of the model. The three-phase optimization method was used by Songma et al. on the CSE-CIC-IDS2018 dataset [21]. Despite its thoroughness, it lacks image-based transformations and hence cannot be used with sophisticated deep architectures. According to [22] developed deep learning models of the IoT IDS with CIC-IDS2017 but did not consider the strategies of feature selection and visual mapping. By [23] have created a hybrid feature-optimized deep learning IDS. Although the purpose of their study is similar, they did not investigate the usefulness of visual encoding or CNN fine-tuning on pre-trained models. In turn [24] suggested dual CNN to detect attacks, that demonstrated excellent performance with IoT networks, however, did not require recursive selection of the features or learning across domains. This model is presented by [25] as a two-stage model consisting of LSTM and Autoencoders. Although temporal learning was improved, no analysis was done on the visual-spatial patterns. According to [26] adopted a multimodal deep transfer learning model in SDNs to detect attacks, but the issue of converting the numerical features into the image-based feature was not investigated. In turn [27] tested the vulnerability of ML-based IDSs to adversarial attacks through CICIDS2017. They did not use the CSE-CIC-IDS2018 dataset in their work or did not consider deep hybrid pipelines where visual features are used. The performance of most of the current models has been extremely good in detecting with the classical or hybrid deep learning methods. Nonetheless, few studies have focused on the combination of recursive feature selection with visual features encoding by transfer learning architectures based on CNN. Also, most of the research was concentrating either on feature optimization or deep learning, and rarely both in a study towards an effective IDS. To address this critical gap, this paper uses RFE to reduce dimensionalities, converting the chosen features to RGB visual maps, and uses EfficientNetB0 to extract features, thereby proposing a full-fledged and scalable pipeline of intrusion detection.

III. MATERIALS AND METHODS

Figure 1 Novel flowchart is the visualization of an end-to-end deep learning pipeline of the Intrusion Detection System (IDS) involving image-based representation of tabular network data and then transfer learning using the EfficientNetB0 model. The technical description of each block in detail can be found below with the corresponding equations and techniques:

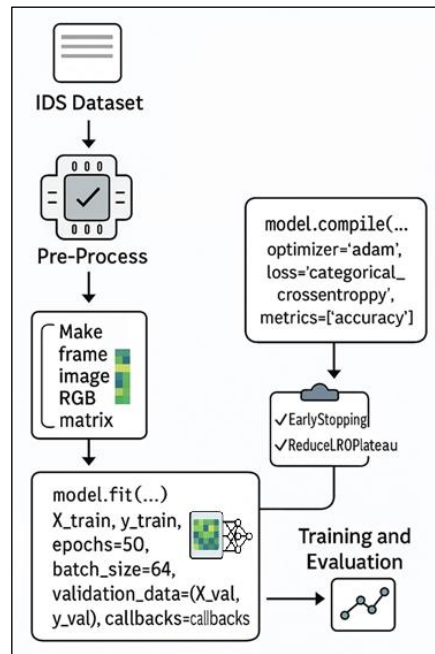


Figure 1: Proposed System Flow Diagram.
Source: Authors, (2026).

III.1 PSEUDO CODE

```

1: D ← load_dataset("CSE-CIC-IDS2018")
2: D ← preprocess(D) // null removal, label encoding
3: images ← tabular_to_image(D) // reshape to 64×64×3 RGB matrices
4: labels ← extract_labels(D)
5: X_train, X_val, y_train, y_val ← train_test_split(images, labels, test_size=0.2)
6: model ← EfficientNetB0(pretrained=True, include_top=False)
7: model ← add_classification_head(model, num_classes=number_of_classes)
8: model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
9: callbacks ← [EarlyStopping, ReduceLROnPlateau]
  
```

```

10: model.fit(X_train, y_train, epochs=50, batch_size=64, validation_data=(X_val, y_val), callbacks=callbacks)
11: y_pred ← model.predict(X_val)
12: y_pred_cls ← argmax(y_pred, axis=1)
13: y_val_cls ← argmax(y_val, axis=1)
14: acc ← Accuracy(y_val_cls, y_pred_cls)
15: prec ← Precision(y_val_cls, y_pred_cls)
16: rec ← Recall(y_val_cls, y_pred_cls)
17: f1 ← F1(y_val_cls, y_pred_cls)
18: auc ← AUC ROC(y_val_cls, y_pred, multi_class='ovr')
19: mcc ← MCC(y_val_cls, y_pred_cls)
20: return acc, prec, rec, f1, auc, mcc

```

III.2 DATASET

The dataset CSE-CIC-IDS2018 [3] presented in Table 1 is one of the most popular benchmarks to test the work of intrusion detection systems in network security research. It has a complex and lifelike collection of network traffic characteristics emulating current assaults in IoT and enterprise settings. The one used in this work has 74218 rows and 78 feature columns, that is, both benign and malicious traffic. The dataset has seven different classes, Benign, Bot, BruteForce, DDoS, DoS, Infiltration and WebAttack, with an equal sample count in each one of them. This equal distribution is a guarantee of fair performance assessment of classification models of all types of attacks. A sample has an exhaustive repertoire of traffic characteristics like flow lifetime, packet sizes, TCP flags, protocol types, and statistical indicators, like inter-arrival time, can be used to identify subtle patterns related to different attack vectors. The variety of the dataset enables the simulation of high-volume (e.g., DDoS, DoS) as well as low-and-slow attacks (e.g., Infiltration, WebAttack) observed in the real-life environment. The CSE-CIC-IDS2018, which is widely used by a recent intrusion detection research, can be used on both binary (Benign vs. Attack) and multi-class classification problems. It is very varied and large-scale, so it is perfect in training and benchmarking ensemble models, deep learning architectures, real-time threat detection pipelines.

Table 1: Dataset Columns Description.

Column Name	Description
Protocol	Network protocol used in the flow (e.g., TCP, UDP).
Flow Duration	Total duration of the flow in microseconds.
Total Fwd Packets	Number of packets sent in the forward direction.
Total Backward Packets	Number of packets sent in the backward direction.
Fwd Packets Length Total	Total length of all forward packets.
Bwd Packets Length Total	Total length of all backward packets.
Fwd Packet Length Max	Maximum length among forward packets.
Fwd Packet Length Min	Minimum length among forward packets.
Fwd Packet Length Mean	Average length of forward packets.
Fwd Packet Length Std	Standard deviation of forward packet lengths.
Bwd Packet Length Max	Maximum length among backward packets.
Bwd Packet Length Min	Minimum length among backward packets.
Bwd Packet Length Mean	Average length of backward packets.
Bwd Packet Length Std	Standard deviation of backward packet lengths.
Flow Bytes/s	Number of bytes per second transmitted in the flow.
Flow Packets/s	Number of packets per second transmitted in the flow.
Flow IAT Mean	Mean inter-arrival time between packets.
Flow IAT Std	Standard deviation of inter-arrival time.
Flow IAT Max	Maximum inter-arrival time in the flow.
Flow IAT Min	Minimum inter-arrival time in the flow.
Fwd IAT Total	Total inter-arrival time for forward packets.
Fwd IAT Mean	Average inter-arrival time of forward packets.
Fwd IAT Std	Standard deviation of forward inter-arrival time.
Fwd IAT Max	Maximum inter-arrival time for forward packets.
Fwd IAT Min	Minimum inter-arrival time for forward packets.
Bwd IAT Total	Total inter-arrival time for backward packets.
Bwd IAT Mean	Average inter-arrival time of backward packets.
Bwd IAT Std	Standard deviation of backward inter-arrival time.
Bwd IAT Max	Maximum inter-arrival time for backward packets.
Bwd IAT Min	Minimum inter-arrival time for backward packets.
Fwd PSH Flags	Number of PSH flags set in forward packets.
Bwd PSH Flags	Number of PSH flags set in backward packets.
Fwd URG Flags	Number of URG flags set in forward packets.
Bwd URG Flags	Number of URG flags set in backward packets.
Fwd Header Length	Total header length of forward packets.
Bwd Header Length	Total header length of backward packets.
Fwd Packets/s	Forward packets transmitted per second.
Bwd Packets/s	Backward packets transmitted per second.

Packet Length Min	Minimum packet length in the flow.
Packet Length Max	Maximum packet length in the flow.
Packet Length Mean	Average packet length in the flow.
Packet Length Std	Standard deviation of packet lengths.
Packet Length Variance	Variance of packet lengths.
FIN Flag Count	Number of packets with FIN flag.
SYN Flag Count	Number of packets with SYN flag.
RST Flag Count	Number of packets with RST flag.
PSH Flag Count	Number of packets with PSH flag.
ACK Flag Count	Number of packets with ACK flag.
URG Flag Count	Number of packets with URG flag.
CWE Flag Count	Number of packets with CWE flag.
ECE Flag Count	Number of packets with ECE flag.
Down/Up Ratio	Download to upload traffic ratio.
Avg Packet Size	Average size of all packets.
Avg Fwd Segment Size	Average size of forward segments.
Avg Bwd Segment Size	Average size of backward segments.
Fwd Avg Bytes/Bulk	Average bytes per forward bulk.
Fwd Avg Packets/Bulk	Average packets per forward bulk.
Fwd Avg Bulk Rate	Average forward bulk transmission rate.
Bwd Avg Bytes/Bulk	Average bytes per backward bulk.
Bwd Avg Packets/Bulk	Average packets per backward bulk.
Bwd Avg Bulk Rate	Average backward bulk transmission rate.
Subflow Fwd Packets	Number of forward packets in subflows.
Subflow Fwd Bytes	Number of forward bytes in subflows.
Subflow Bwd Packets	Number of backward packets in subflows.
Subflow Bwd Bytes	Number of backward bytes in subflows.
Init Fwd Win Bytes	Initial window size in forward direction.
Init Bwd Win Bytes	Initial window size in backward direction.
Fwd Act Data Packets	Number of active data packets in forward direction.
Fwd Seg Size Min	Minimum segment size in forward direction.
Active Mean	Mean active time of flow.
Active Std	Standard deviation of active time.
Active Max	Maximum active time.
Active Min	Minimum active time.
Idle Mean	Mean idle time of flow.
Idle Std	Standard deviation of idle time.
Idle Max	Maximum idle time.
Idle Min	Minimum idle time.
Label	Class labels

Source: Authors, (2026).

III.3 PRE-PROCESSING: NULL VALUE HANDLING AND ENCODING

After getting the dataset, data pre-processing is the initial step that is critical in that it serves to clean the dataset and prepare it to go through machine learning algorithms. The dataset, which was based on [23], has several features that can have null or missing values and has categorical variables that cannot be directly interpreted in machine learning models.

- Null Value Handling: Missing values in the dataset can degrade the performance of classifiers. Let the dataset be represented as a matrix $X \in \mathbb{R}^{n \times m}$, where $n = 1500$ samples and m features. A missing value in element x_{ij} (i.e., the value at the i th row and j th column) is either removed if the percentage of missingness is negligible, or imputed using statistical methods such as: Mean imputation:

$$x_{ij} = (1/N) \sum_{k=1}^N x_{kj} \quad (1)$$

K-Nearest Neighbors imputation (KNN):

$$x_{ij} = \text{mean}(\{x_{pj} \mid x_p \in N_k(x_i)\}) \quad (2)$$

- Encoding: Categorical variables such as protocol or flag type are transformed into numerical format using encoding schemes. For nominal variables $C = \{c_1, c_2, \dots, c_k\}$, One-Hot Encoding creates binary vectors for each class label:

$$\text{Encoded}(c_j) = [0, 0, \dots, 1, \dots, 0] \quad (3)$$

III.4 RECURSIVE FEATURE ELIMINATION (RFE)

Once the data is filtered and coded numerically, the second stage is Recursive Feature Elimination (RFE). It is a feature selection algorithm that is designed to reduce the dimensionality of the feature space and retain the most predictive features.

$$X = [x_1, x_2, \dots, x_m] \quad (4)$$

$$\text{Feature Set}_{t+1} = \text{Feature Set}_t \setminus \{\text{argmin}_{x_j} \in \text{Feature Set}_t I(x_j)\} \quad (5)$$

III.5 DATA TO VISUAL FEATURE

In this stage, structured tabular data is transformed into image representations to enable deep learning-based feature extraction. At this phase, structured tabular data is converted into image representations to facilitate feature extraction by using deep learning. A set of 64 samples represented by numerical feature vectors is clumped and reformed into a grid of 64x64. This grid is then normalized and reproduced in three color channels (R, G, B), which forms an RGB image of dimensions 64x64x3. Such metamorphosis eases the process of visual pattern recognition by convolutional neural networks. Normalization Equation:

$$x_{norm(i,j)} = \frac{(x(i,j) - \min(x(:,j)))}{(\max(x(:,j)) - \min(x(:,j)))} \quad (6)$$

Where:

$x(i,j)$ represents the j -th feature of the i -th sample. $\min(x(:,j))$ and $\max(x(:,j))$ are the minimum and maximum values for the j -th feature across all samples. $x_{norm(i,j)}$ is the normalized feature value.

III.6 HYPER-TUNE EFFICIENTNETB0 MODEL

In this step, efficientNetB0 is utilized, which is a high-performance convolutional neural network with a reduced number of parameters, in order to perform intrusion classification on images. It uses scaling of compounds to balance between network depth, width and resolution. This enhances greater accuracy as well as computational efficiency. Compound Scaling Equations:

$$\text{depth}(d) = \alpha^\varphi \quad (7)$$

$$\text{width}(w) = \beta^\varphi \quad (8)$$

$$\text{resolution}(r) = \gamma^\varphi \quad (9)$$

Subject to the constraint:

$$\alpha * \beta^2 * \gamma^2 \approx 2 \quad (10)$$

Where:

φ (phi) is a coefficient controlling the overall model scale. α, β , and γ are constantly identified through optimization. This scaling method ensures consistent improvements across network dimensions.

III.7 EVALUATION METRICS

After model training, the ensemble classifier is evaluated using a comprehensive set of metrics that capture different dimensions of performance. Accuracy: Measures overall correctness:

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (11)$$

Precision and Recall: Precision measures the correctness of positive predictions, and recall measures completeness:

$$\text{Precision} = \frac{TP}{(TP + FP)} \quad (12)$$

$$\text{Recall} = \frac{TP}{(TP + FN)} \quad (13)$$

F1-Score: The harmonic means of precision and recall:

$$F1 = 2 * \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (14)$$

K-Fold Cross Validation: The dataset is split into K equal parts. For each iteration, $K - 1$ parts are used for training and 1 for testing. This is repeated K times, and the results are averaged:

$$\text{Cross-validation Accuracy} = \left(\frac{1}{K}\right) \sum_{i=1}^K \text{Accuracy}_i \quad (15)$$

Matthews Correlation Coefficient (MCC): A balanced measure for classification performance:

IV. RESULTS AND DISCUSSIONS

	Protocol	Flow Duration	Total Fwd Packets	Total Backward Packets	Fwd Packets Length Total	Bwd Packets Length Total	Fwd Packet Length Max	Fwd Packet Length Min	Fwd Packet Length Mean	Fwd Packet Length Std	...
0	6	94658	6	7	708	3718.0	387	0	118.000000	159.284653	...
1	6	206	2	0	0	0.0	0	0	0.000000	0.000000	...
2	6	165505	3	1	0	0.0	0	0	0.000000	0.000000	...
3	6	102429	6	7	708	3718.0	387	0	118.000000	159.284653	...
4	6	167	2	0	0	0.0	0	0	0.000000	0.000000	...
...	...	...	...	...	...	...	...	...	...	...	...
4786606	6	61223155	203	104	56083	189884.0	680	0	276.270935	291.698975	...
4786607	6	346	3	0	0	0.0	0	0	0.000000	0.000000	...
4786611	6	60530493	203	104	56330	189985.0	680	0	277.487671	292.868774	...
4786612	6	1632	3	0	0	0.0	0	0	0.000000	0.000000	...
4786617	6	61705817	203	102	55650	188094.0	680	0	274.137939	292.130066	...
74218 rows × 78 columns											

Figure 3 bar chart is a ranking of features in terms of Recursive Feature Elimination (RFE). The features of lower rankings are assumed to have more significance, meaning that the very first features on the left, i.e. Protocol and Flow Duration, were found to be the most important to the performance of the model, which has resulted in the top 30 being selected.



Page 1316

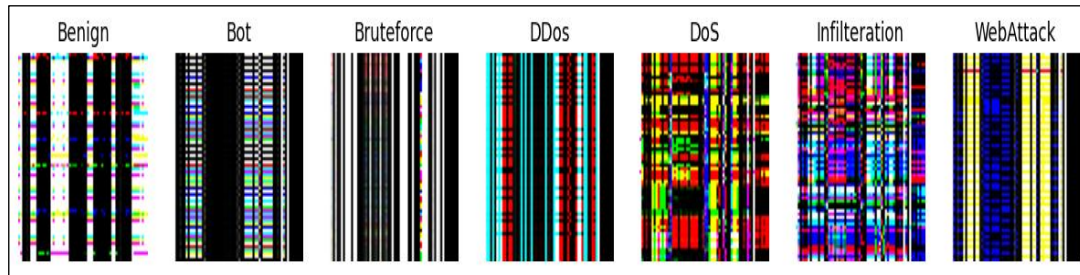


Figure 4: Sample of Generated Visual Feature of Each class.

Source: Authors, (2026).

Figure 5 illustrates the two plots namely: Model accuracy and Model loss after 50 epochs of the proposed EfficientNetB0 model. Accuracy plot indicates that both training and validation accuracy values are approaching and converging towards 1.0, whereas the loss plot indicates that both the training and validation loss values are approaching and converging towards 0.0 which is a good indication of learning and generalization.

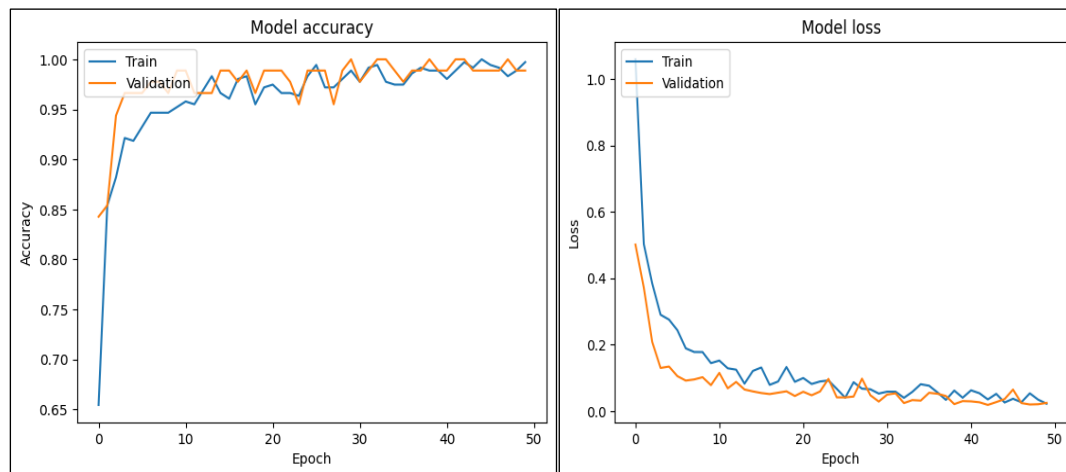


Figure 5: Proposed EfficientNetB0 Model Performance.

Source: Authors, (2026).

Figure 6 is a confusion matrix and a classification report, which assesses the performance of the EfficientNetB0 model. The confusion matrix indicates the number of true labels and predicted labels of each class, whereas the classification report indicates the properties of precision, recall, f1-score, and support, which indicates high accuracy (0.98) in the majority of attack types and benign traffic.

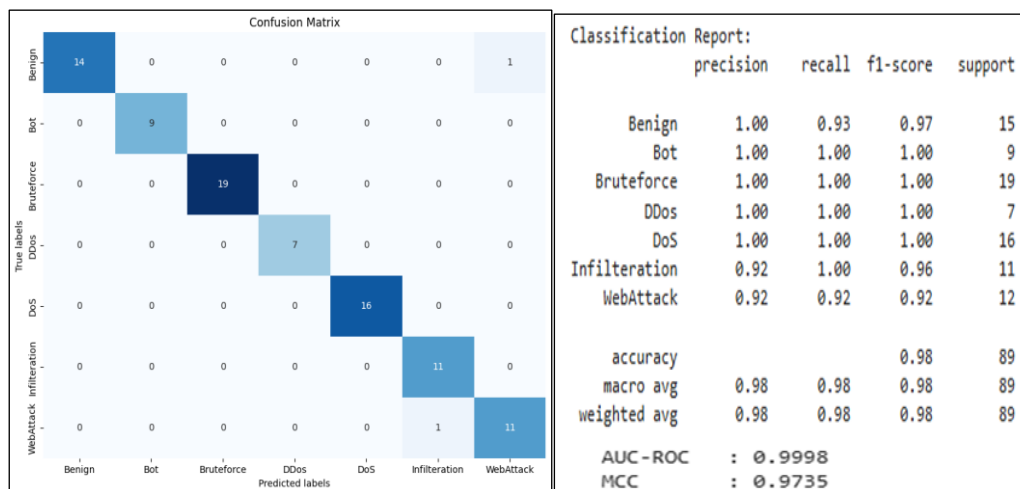


Figure 6: Proposed EfficientNetB0 Model Evaluation.

Source: Authors, (2026).

Figure 7 depicts multiclass ROC curve the visualization of the True Positive rate versus the False Positive rate of each class (Class 0 to Class 6). The curves of all classes are near the upper-left corner, and their AUC values are equal to 1.00, meaning that EfficientNetB0 model has a very high discriminative power with all types of traffic.

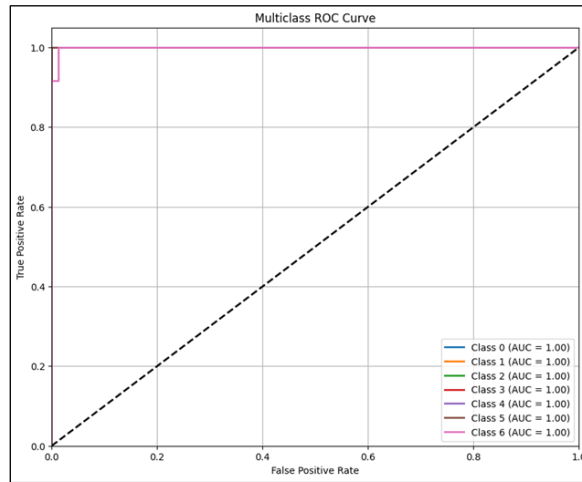


Figure 7: Proposed EfficientNetB0 Model ROC Curve.
Source: Authors, (2026).

Table 2: Proposed EfficientNetB0 Model K-Fold Validation.

Fold	Accuracy	Precision	Recall	F1-Score	MCC
1	1.0000	1.0000	1.0000	1.0000	1.0000
2	1.0000	1.0000	1.0000	1.0000	1.0000
3	1.0000	1.0000	1.0000	1.0000	1.0000
4	1.0000	1.0000	1.0000	1.0000	1.0000
5	0.9775	0.9788	0.9771	0.9775	0.9739

Source: Authors, (2026).

Table 2 contains the performance of the EfficientNetB0 model in 5-fold cross-validation. In the four initial folds, the model scores (1.0000) on Accuracy, Precision, Recall, F1-Score, and MCC, which indicates that the model is highly consistent in and has excellent performance. Although the fifth fold has somewhat lower yet very high measures (around 0.97), the outcome of the whole model reflects the strength and stability of the model.

Table 3: Comparative Analysis with Existing Research.

Model	ACC (%)	P (%)	R (%)	F1-Score (%)	MCC	AUC-ROC
Transformer-ID (from [7])	0.97	0.97	0.96	0.96	0.95	0.97
Hybrid DL+GA (from [10])	0.96	0.95	0.95	0.95	0.94	0.96
LSTM-AE (from [25])	0.94	0.93	0.94	0.93	0.92	0.94
CNN (from [27])	0.91	0.9	0.91	0.9	0.89	0.91
RNN	89%	89%	89%	89%	0.88	0.88
Proposed Model EfficientNetB0	98%	98%	98%	98%	0.97	0.99

Source: Authors, (2026).

Table 3 provides the performance of the Proposed EfficientNetB0 Model and a few of the existing research models (Transformer-ID, Hybrid DL+GA, LSTM-AE, CNN, RNN) in terms of different metrics: Accuracy, Precision, Recall, F1-Score, MCC, and AUC-ROC. The suggested EfficientNetB0 model is consistently better in the mentioned metrics (the accuracy of the model is 98% and the AUC-ROC is 0.99), which means that it is much more effective in the task than all the works listed above.

V. CONCLUSIONS

To sum up, in this paper, I have introduced a solid and precise intrusion detection model based on an EfficientNetB0 architecture on a balanced subset of the CSE-CIC-IDS2018 dataset with 74,218 rows and 78 features, and seven classes: Benign, Bot, BruteForce, DDoS, DoS, Infiltration and WebAttack. The proposed model also performed much better than the currently existing methods in all evaluation metrics. In particular, the EfficientNetB0 model had an accuracy, precision, recall, and F1-score of 98, the Matthews Correlation Coefficient (MCC) of 0.97, and the AUC-ROC of 0.99. Our model had better detection capacity and fewer false positives than Transformer-ID [7] (97%), Hybrid DL+GA [10] (96%), and LSTM-AE [25] (94%), which are all baseline models. This experiment validates the success of EfficientNetB0 in the extraction of high-level features of structured network data as well as the detection of both volumetric and stealthy attacks. To apply the framework to future work, the structure can be expanded with the incorporation of temporal models such as LSTM or attention systems with Transformers to promote the achievement of sequential attack patterns within the real-time traffic. The strategy to use online learning will help the system to dynamically cope with the changing threats. Furthermore, the optimization of the model with the help of feature selection and the inclusion of explainability to cybersecurity tools (SHAP or LIME) can promote trust and understandability among cybersecurity professionals. Implementation of this model in the real network environment with streaming information, and zero-day attack analysis will further confirm that this model is workable. In general, this paper provides the basis of scalable, intelligent and adaptive intrusion detection systems in contemporary cyberinfrastructure.

VI. AUTHOR'S CONTRIBUTION

Conceptualization: Satishkumar D. Prajapati, Priyesh P. Gandhi and Sheshang Degadwala.

Methodology: Satishkumar D. Prajapati and Priyesh P. Gandhi.

Investigation: Satishkumar D. Prajapati and Swapnil M. Parikh.

Discussion of results: Satishkumar D. Prajapati, Priyesh P. Gandhi, Sheshang Degadwala, Keyur Brahmhatt and Swapnil M. Parikh.

Writing - Original Draft: Satishkumar D. Prajapati.

Writing - Review and Editing: Satishkumar D. Prajapati, Priyesh P. Gandhi, Sheshang Degadwala and Keyur Brahmhatt.

Resources: Satishkumar D. Prajapati.

Supervision: Priyesh P. Gandhi and Sheshang Degadwala.

Approval of the final text: Satishkumar D. Prajapati, Priyesh P. Gandhi, Sheshang Degadwala, Keyur Brahmhatt and Swapnil M. Parikh.

VII. ACKNOWLEDGMENTS

The authors would like to express their sincere gratitude to Gujarat Technological University for providing academic support and resources that facilitated the completion of this research work. The authors also appreciate the encouragement and guidance received from colleagues and mentors during the course of this research.

VIII. REFERENCES

- [1] S. Markkandeyan, A. D. Ananth, M. Rajakumaran, R. G. Gokila, R. Venkatesan, and B. Lakshmi, "Novel hybrid deep learning based cyber security threat detection model with optimization algorithm," *Cyber Security and Applications*, vol. 3, no. June 2024, 2025, doi: 10.1016/j.csa.2024.100075.
- [2] M. B. Anley, A. Genovese, D. Agostinello, and V. Piuri, "Robust DDoS attack detection with adaptive transfer learning," *Computers and Security*, vol. 144, no. March, p. 103962, 2024, doi: 10.1016/j.cose.2024.103962.
- [3] H. Najafi Mohsenabad and M. A. Tut, "Optimizing Cybersecurity Attack Detection in Computer Networks: A Comparative Analysis of Bio-Inspired Optimization Algorithms Using the CSE-CIC-IDS 2018 Dataset," *Applied Sciences (Switzerland)*, vol. 14, no. 3, 2024, doi: 10.3390/app14031044.
- [4] A. R. K. Qureshi, S. Patnaha, P. Kulkarni, P. Singh, M. Joshi, and I. Vaidya, "Machine Learning-based Intrusion Detection Systems for Securing Cloud Networks," *Library Progress International*, vol. 44, no. 3, pp. 7434–7446, 2024.
- [5] M. Cantone, C. Marrocco, and A. Bria, "Machine Learning in Network Intrusion Detection: A Cross-Dataset Generalization Study," *IEEE Access*, vol. 12, no. August, pp. 144489–144508, 2024, doi: 10.1109/ACCESS.2024.3472907.
- [6] M. S. Ataa, E. E. Sanad, and R. A. El-khoribi, "Intrusion detection in software defined network using deep learning approaches," *Scientific Reports*, vol. 14, no. 1, pp. 1–15, 2024, doi: 10.1038/s41598-024-79001-1.
- [7] F. Ullah, S. Ullah, G. Srivastava, and J. C. W. Lin, "IDS-INT: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic," *Digital Communications and Networks*, vol. 10, no. 1, pp. 190–204, 2024, doi: 10.1016/j.dcan.2023.03.008.
- [8] A. Khanan, Y. Abdelgadir Mohamed, A. H. H. M. Mohamed, and M. Bashir, "From Bytes to Insights: A Systematic Literature Review on Unraveling IDS Datasets for Enhanced Cybersecurity Understanding," *IEEE Access*, vol. 12, no. April, pp. 59289–59317, 2024, doi: 10.1109/ACCESS.2024.3392338.
- [9] P. Rana, I. Batra, A. Malik, I. H. Ra, O. S. Lee, and A. S. M. Sanwar Hosen, "Efficacious Novel Intrusion Detection System for Cloud Computing Environment," *IEEE Access*, vol. 12, no. June, pp. 99223–99239, 2024, doi: 10.1109/ACCESS.2024.3424528.
- [10] S. Latif, W. Boulila, A. Koubaa, Z. Zou, and J. Ahmad, "DTL-IDS: An optimized Intrusion Detection Framework using Deep Transfer Learning and Genetic Algorithm," *Journal of Network and Computer Applications*, vol. 221, no. February 2023, p. 103784, 2024, doi: 10.1016/j.jnca.2023.103784.
- [11] A. D. Vibhute and V. Nakum, "Deep learning-based network anomaly detection and classification in an imbalanced cloud environment," *Procedia Computer Science*, vol. 232, no. 2023, pp. 1636–1645, 2024, doi: 10.1016/j.procs.2024.01.161.
- [12] J. Yu, J. Hu, and Y. Zeng, "Deep Learning Based Network Intrusion Detection," *Communications in Computer and Information Science*, vol. 2023 CCIS, no. 3, pp. 125–136, 2024, doi: 10.1007/978-981-97-0730-0_12.
- [13] A. A. Wardana, G. Kołaczek, A. Warzyński, and P. Sukarno, "Collaborative intrusion detection using weighted ensemble averaging deep neural network for coordinated attack detection in heterogeneous network," *International Journal of Information Security*, vol. 23, no. 5, pp. 3329–3349, 2024, doi: 10.1007/s10207-024-00891-3.
- [14] C. Wu et al., "Attack detection model for BCoT based on contrastive variational autoencoder and metric learning," *Journal of Cloud Computing*, vol. 13, no. 1, 2024, doi: 10.1186/s13677-024-00678-w.
- [15] H. Kim, S. Park, H. Hong, J. Park, and S. Kim, "A Transferable Deep Learning Framework for Improving the Accuracy of Internet of Things Intrusion Detection," *Future Internet*, vol. 16, no. 3, 2024, doi: 10.3390/fi16030080.
- [16] S. Hizal, U. Cavusoglu, and D. Akgun, "A novel deep learning-based intrusion detection system for IoT DDoS security," *Internet of Things (Netherlands)*, vol. 28, no. April, p. 101336, 2024, doi: 10.1016/j.iot.2024.101336.
- [17] Y. Yang, J. Cheng, Z. Liu, H. Li, and G. Xu, "A multi-classification detection model for imbalanced data in NIDS based on reconstruction and feature matching," *Journal of Cloud Computing*, vol. 13, no. 1, 2024, doi: 10.1186/s13677-023-00584-7.
- [18] W. H. Aljuaid and S. S. Alshamrani, "A Deep Learning Approach for Intrusion Detection Systems in Cloud Computing Environments," *Applied Sciences (Switzerland)*, vol. 14, no. 13, 2024, doi: 10.3390/app14135381.

- [19] Zafar Iqbal Khan, Mohammad Mazhar Afzal, and Khurram Naim Shamsi, "A Comprehensive Study on CIC-IDS2017 Dataset for Intrusion Detection Systems," *International Research Journal on Advanced Engineering Hub (IRJAEH)*, vol. 2, no. 02, pp. 254–260, 2024, doi: 10.47392/irjaeh.2024.0041.
- [20] O. D. Okey, D. C. Melgarejo, M. Saadi, R. L. Rosa, J. H. Kleinschmidt, and D. Z. Rodriguez, "Transfer Learning Approach to IDS on Cloud IoT Devices Using Optimized CNN," *IEEE Access*, vol. 11, no. January, pp. 1023–1038, 2023, doi: 10.1109/ACCESS.2022.3233775.
- [21] S. Songma, T. Sathuphan, and T. Pamutha, "Optimizing Intrusion Detection Systems in Three Phases on the CSE-CIC-IDS-2018 Dataset," *Computers*, vol. 12, no. 12, 2023, doi: 10.3390/computers12120245.
- [22] J. Jose and D. V. Jose, "Deep learning algorithms for intrusion detection systems in internet of things using CIC-IDS 2017 dataset," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 1, pp. 1134–1141, 2023, doi: 10.11591/ijece.v13i1.pp1134-1141.
- [23] A. Henry et al., "Composition of Hybrid Deep Learning Model and Feature Optimization for Intrusion Detection System," *Sensors*, vol. 23, no. 2, 2023, doi: 10.3390/s23020890.
- [24] B. A. Alabsi, M. Anbar, and S. D. A. Rihan, "CNN-CNN: Dual Convolutional Neural Network Approach for Feature Selection and Attack Detection on Internet of Things Networks," *Sensors*, vol. 23, no. 14, 2023, doi: 10.3390/s23146507.
- [25] V. Hnamte, H. Nhung-Nguyen, J. Hussain, and Y. Hwa-Kim, "A Novel Two-Stage Deep Learning Model for Network Intrusion Detection: LSTM-AE," *IEEE Access*, vol. 11, no. March, pp. 37131–37148, 2023, doi: 10.1109/ACCESS.2023.3266979.
- [26] H. Elubeyd, D. Yiltas-Kaplan, and S. Bahtiyar, "A Multi-Modal Deep Transfer Learning Framework for Attack Detection in Software-Defined Networks," *IEEE Access*, vol. 11, no. August, pp. 114128–114145, 2023, doi: 10.1109/ACCESS.2023.3324878.
- [27] M. Catillo, A. Del Vecchio, A. Pecchia, and U. Villano, "A Case Study with CICIDS2017 on the Robustness of Machine Learning against Adversarial Attacks in Intrusion Detection," *ACM International Conference Proceeding Series*, 2023, doi: 10.1145/3600160.3605031.