

Article

An Explainable Hybrid Feature Selection Framework for Deep Learning-Based Intrusion Detection

Martins Onyekwelu Onuorah ^{1,2,*} , Yanxia Sun ¹  and Daniel Mashao ³¹ Department of Electrical and Electronic Engineering Science, University of Johannesburg, Auckland Park 2006, South Africa² Department of Computer Science, Kigali Independent University, Kigali P.O. Box 2280, Rwanda³ Faculty of Engineering and Built Environment, University of Johannesburg, Auckland Park 2006, South Africa

* Correspondence: martins.onuorah@ul.ac.rw

Abstract

High-dimensional network traffic data can increase the computational cost of intrusion detection systems, introduce redundant information, and reduce model interpretability. Severe class imbalance can further impair the detection of minority attack classes. This study proposes IG²AFS-DNN, an explainable hybrid feature selection framework that combines information gain filtering with a deep neural network-guided genetic algorithm. Information gain first removes weakly relevant features, after which the genetic algorithm identifies a compact feature subset by jointly considering macro F1 and dimensionality reduction. A deep neural network performs multiclass intrusion detection, while SHAP provides global and local post hoc explanations of model predictions. The framework was evaluated on the CSE-CIC-IDS2018 and TII-SSRC-23 datasets. It reduced the CSE-CIC-IDS2018 feature set from 80 to 23 features (a 71.25% reduction) and the TII-SSRC-23 feature set from 86 to 15 features (an 82.56% reduction). The resulting models achieved macro F1-scores of 0.833 and 0.992 and Matthews correlation coefficients of 0.972 and 0.999 on CSE-CIC-IDS2018 and TII-SSRC-23, respectively. Ablation and baseline comparisons showed that the complete hybrid framework outperformed the full-feature DNN and conventional linear and probabilistic classifiers while remaining competitive with tree-based ensemble models. SHAP analysis indicated that the selected features captured meaningful packet, flow, timing, and protocol characteristics. These findings demonstrate that IG²AFS-DNN provides a favorable balance among feature compactness, multiclass detection performance, and model interpretability across two heterogeneous intrusion detection datasets.

Keywords: intrusion detection; hybrid feature selection; information gain; genetic algorithm; deep neural network; explainable artificial intelligence; SHAP; class imbalance



Academic Editor: Thomas Lindner

Received: 18 September 2026

Revised: 19 October 2026

Accepted: 20 October 2026

Published: 27 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Intrusion detection systems (IDS) play a critical role in modern cybersecurity infrastructure, particularly amid the rapid expansion of cloud computing, the Internet of Things (IoT), and distributed network environments [1]. The increasing sophistication and frequency of cyberattacks have significantly expanded the digital attack surface, exposing organizations to evolving threats driven by financial incentives, advanced attack tools, and persistent vulnerabilities [2,3]. The borderless and anonymous nature of cyber threats further complicates detection and mitigation efforts, thereby necessitating intelligent and adaptive security mechanisms [4,5].

Traditional IDS approaches, including signature-based and rule-based systems such as Snort [6] and Suricata [7], are effective in detecting known threats but struggle to identify zero-day attacks and dynamic intrusion patterns. As a result, machine learning (ML) and deep learning (DL) techniques have been increasingly adopted to enhance intrusion detection using automated pattern recognition and adaptive learning [8].

Deep learning models, including deep neural networks (DNNs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs), have demonstrated strong capabilities in modeling high-dimensional network traffic data and improving detection accuracy across diverse attack types [8]; however, several challenges remain [9,10]. First, high-dimensional datasets often contain redundant and irrelevant features, increasing computational complexity and the risk of overfitting. Second, deep learning models are computationally intensive, limiting their scalability in resource-constrained environments [11]. Third, the black-box nature of these models reduces interpretability, which is essential in security-critical domains.

Feature selection has been widely employed to address the challenges of high-dimensional data. Filter-based methods, such as information gain (IG), provide efficient feature ranking, while wrapper-based approaches, such as genetic algorithms (GA), enable classifier-driven feature optimization [12,13]. Hybrid feature selection strategies that combine filter and wrapper methods have demonstrated improved performance and computational efficiency. However, many existing approaches rely on classical machine learning models or do not integrate explainable artificial intelligence (XAI) techniques.

Although previous studies, including CSE-CIC-IDS2018 [14] and TII-SSRC-23 [15], have demonstrated the benefits of feature selection, deep learning, and explainable artificial intelligence for intrusion detection, several limitations remain. Many feature selection studies [16,17] use either filter or wrapper methods in isolation, while hybrid approaches frequently rely on conventional machine learning classifiers. Studies incorporating explainability often apply it to full-feature models without examining whether the selected feature subsets retain meaningful network traffic characteristics. Furthermore, limited attention has been given to the stability of stochastic feature selection, the individual contribution of each selection stage and validation across multiple contemporary intrusion detection datasets. These limitations motivate a framework that combines efficient relevance filtering, classifier-guided subset optimization, controlled ablation, and post hoc explanation.

To address these limitations, this study proposes IG²AFS-DNN, a hybrid feature selection framework that combines information gain filtering with a DNN-guided genetic algorithm. Information gain initially removes weakly relevant features, thereby reducing the search space available to the genetic algorithm. The genetic algorithm then identifies a compact subset by optimizing a fitness function that prioritizes macro F1 while rewarding dimensionality reduction. The selected features are used to train a multiclass DNN, and SHAP is subsequently applied to provide global and instance-level explanations. The framework balances predictive performance, feature compactness, and interpretability and is evaluated independently on the CSE-CIC-IDS2018 and TII-SSRC-23 datasets. The principal contributions of this study are as follows:

1. A sequential hybrid feature selection framework that uses information gain to reduce the initial search space and a DNN-guided genetic algorithm to identify compact feature subsets using a fitness function that jointly considers macro F1 and dimensionality reduction.
2. A stability-aware GA evaluation procedure that compares multiple population and generation configurations across repeated runs using predictive performance, feature count, Jaccard stability, and computational cost.

3. A controlled ablation study that evaluates the incremental contributions of information gain filtering and GA-based optimization by comparing the full-feature DNN, IG-DNN and the complete IG²AFS-DNN framework under consistent experimental conditions.
4. A multi-dataset evaluation of CSE-CIC-IDS2018 and TII-SSRC-23, supported by class-specific metrics, Matthews correlation coefficient, bootstrap confidence intervals, paired McNemar's tests, and comparisons with conventional machine learning baselines.
5. Post hoc global and local explainability using SHAP, enabling examination of how the selected packet, flow, timing, and protocol features influence the predictions of the DNN.

2. Literature Review

Early deep learning intrusion detection systems (DL-IDSs) established the feasibility of automated, high-accuracy intrusion detection but also exposed important gaps. Vinayakumar et al. [17] demonstrated a multi-dataset DNN that generalizes across KDDCup'99, NSL-KDD, UNSW-NB15, Kyoto, WSN-DS, and CICIDS2017, with careful hyperparameter tuning and strong ROC/precision–recall performance; however, they also acknowledged that older, anonymized datasets (e.g., KDDCup'99) lack realism and limit external validity. Kasongo and Sun [12] applied a wrapper-based feature extraction unit with a feed-forward DNN to UNSW-NB15 and AWID, achieving strong accuracies (binary/multiclass). However, considerable training/validation costs were noted, highlighting the computational burden of DL-IDS pipelines. Sonal and Deswal [9] stacked LSTM, GRU, RNN, and CNN for IoT gateway security on CIC-IDS2017 and surpassed prior baselines (>99% across metrics). However, the study relied on a single, aging dataset and did not employ feature selection or explainability, leaving a black-box, resource-heavy model. These DL-IDS works validate the power of deep models but motivate (i) movement to modern benchmarks, (ii) principled feature selection to reduce cost, and (iii) explainable AI (XAI) to improve trust.

Recent studies increasingly target contemporary datasets to capture evolving attack surfaces. Zhao et al. [18] combined CFS-DE feature selection with a stacked ensemble and achieved up to 99.87% accuracy/99.88% F1 on CSE-CIC-IDS2018, showing how dimensionality reduction can both speed training and lift accuracy; however, their models remained classical ML and non-interpretable. Vibhute and Nakuma [19] coupled random forest-based selection with a CNN on CSE-CIC-IDS2018, selecting 17/80 features and reaching 97% test accuracy. However, the study did not test cross-dataset generalization to TII-SSRC-23 or add XAI. Rani et al. [16] blended fuzzy rules with GAN augmentation (GFS-GAN) and reported 99% accuracy on TII-SSRC-23 and NSL-KDD; despite interpretability using fuzzy rules, computational cost and limited benchmarking remained concerns. Ciaramella et al. [20] advanced explainable DL for critical infrastructure by converting PCAPs into images and testing 10 CNNs on an IEC-104 dataset and the TII-SSRC-23; they used Grad-CAM++/Score-CAM for transparency but did not incorporate feature selection and experienced potential overfitting on smaller sample sizes. Modern datasets (especially CSE-CIC-IDS2018 and TII-SSRC-23) are becoming standard. The best results often pair them with dimensionality reduction or XAI, but rarely both together with deep models and cross-dataset validation.

Hybrid feature selection (FS) consistently improves IDS efficiency and, often, accuracy. Tirumanadham et al. [21] introduced a two-tier BR2-2T pipeline (Boruta + Ridge) with robust preprocessing (MICE, SMOTE, normalization) and a tiered ensemble. Impressive accuracies in prediction underscore the general utility of hybrid FS; however, the work does not test complex network traffic data or DL. Yin et al. [8] proposed IGRF-RFE (IG + RF importance + RFE) for MLP on UNSW-NB15, trimming 42 features to 23 and modestly improving accuracy (4%). This study solidly demonstrated that filter + wrapper reduces redundancy and cost, albeit on an older dataset without XAI. In IoT-IDS,

Ayad et al. [22] fused correlation-based filters (Pearson/Spearman) with RFE (DT/RF), cutting BoT-IoT (46vto 12), TON-IoT (45→15), and CIC-DDoS2019 (79→35), reaching near-perfect ML performance with very low detection times; scalability to high-dimensional DL and explainability were not addressed. Bakır and Ceviz [13] combined mutual information + sequential FS with GA hyperparameter tuning on CICIDS2017, slashing 80→6 features and achieving ~99.9% accuracy/F1 with XGBoost/RF; however, the study remained in the classical ML/older-dataset regime. Hybrid FS (filter + wrapper) is repeatedly effective for dimension reduction and speed, yet most studies (i) use classical ML, (ii) overlook modern datasets in combination, and (iii) omit XAI, leaving room for DL-integrated hybrid FS with interpretability.

Explainability is moving from an add-on visualization to a core capability. Wali et al. [23] used SHAP not only for interpretation but as a credibility defense. A CAM flags predictions whose local vs. global SHAP attributions disagree, reducing adversarial evasion on CSE-CIC-IDS2018 and CIC-IoT-23 to ~1.9% while keeping F1 $\approx$ 0.99; the dual-pipeline design is powerful but resource heavy and not FS aware. Fatema et al. [24] proposed FedXAIIDS, a federated ANN with global SHAP ranking on CICIoT2023 to preserve privacy and interpretability ($\approx$ 88% accuracy). However, accuracy trails centralized DL, and no hybrid FS was applied. Alatawi [25] combined ensemble/transfer learning with LIME/SHAP across merged legacy datasets to reach $\approx$ 95% accuracy and reduce training time, but again without hybrid FS and on older data. Hariharan et al. [26] systematically compared global/local XAI (PI, SHAP, LIME, CIU) for RF/XGBoost/LightGBM on NSL-KDD/Kaggle IDS, showing that top-k features selected by SHAP/PI can improve accuracy; however, the analysis is constrained by classical ML and legacy datasets. XAI demonstrably boosts trust and can even harden defenses, but most IDS-XAI studies (i) omit hybrid FS, (ii) rely on non-DL models or older datasets, or (iii) add computational overhead without profiling.

Across these four streams, clear patterns emerge. Deep learning delivers superior detection but can be computationally expensive and opaque without FS and XAI. Studies on modern benchmarks (CSE-CIC-IDS2018, TII-SSRC-23) typically optimize either efficiency (using FS) or transparency (using XAI), but rarely integrate both within a deep model and validate across datasets. Hybrid feature selection (filter + wrapper) consistently proves effective, yet most pipelines stop at classical ML and do not tightly integrate FS with DL training. XAI adds trust and sometimes robustness but is commonly post hoc and not coupled with the FS pipeline or resource profiling.

Existing IDS studies have examined filter and wrapper selection, deep classifiers, and post hoc explanation, but the reviewed work offers limited evidence on the combined effects of sequential information gain and DNN-guided GA selection, the stability of the selected subsets, and the contribution of each selection stage. This study evaluates these aspects independently on two intrusion-detection datasets and examines the fitted models using SHAP.

3. Materials and Methods

This section describes the datasets, preprocessing procedures, hybrid feature selection framework, deep learning model, and explainable AI approach employed in this study. The schematic diagram of the proposed approach is given in Figure 1.

3.1. Datasets

This study used two contemporary intrusion detection datasets to evaluate the proposed framework: CSE-CIC-IDS2018 [13] and TII-SSRC-23 [14]. These datasets realistically represent modern network environments and include both benign traffic and diverse attack

types. The CSE-CIC-IDS2018 dataset includes more than 80 flow-based features that capture statistical, temporal, and behavioral characteristics of network traffic. It encompasses a wide range of attacks, including distributed denial-of-service (DDoS), brute force, botnet, and infiltration attacks. The TII-SSRC-23 dataset reflects modern cyber-physical and IoT environments, with 86 features describing network flows. It categorizes traffic into benign and multiple malicious classes, including brute force, denial-of-service, and botnet-related activities.

IG²AFS-DNN methodology

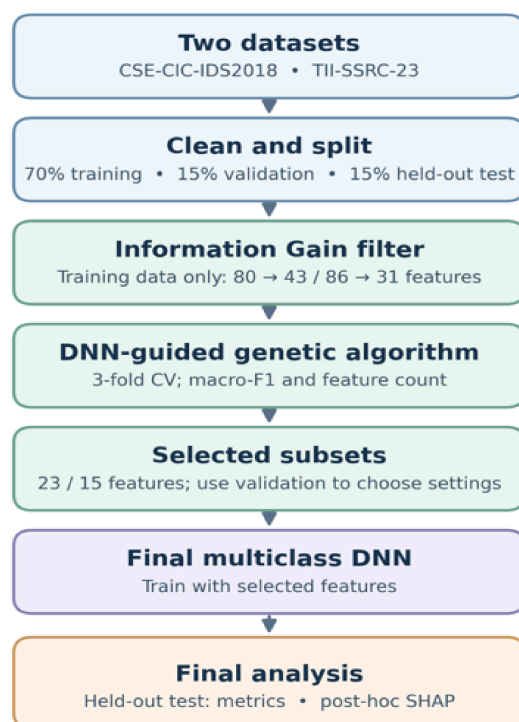


Figure 1. Schematic diagram of the proposed approach.

3.2. Data Preprocessing

The datasets were first inspected for missing, infinite, and inconsistent values. Missing and infinite values were replaced with zeros or removed where appropriate to ensure data integrity. We also identified and removed duplicate records. Categorical features, where present, were transformed using label encoding. Numerical features were normalized using min–max scaling to map values into the range [0, 1]. The normalization parameters were computed on the training set and then applied to the validation and test sets to prevent data leakage.

The datasets were partitioned into training (70%), validation (15%), and test (15%) sets using stratified sampling to preserve class distributions. A fixed random seed of 42 was used for data partitioning to ensure consistent and reproducible allocation of observations across the three subsets. Feature selection was performed exclusively on the training data, and the selected feature subsets were subsequently applied to the validation and test sets. No synthetic oversampling technique was applied in this study to evaluate the robustness of the proposed framework under realistic class imbalance conditions.

3.3. Hybrid Feature Selection Framework

The proposed IG²AFS-DNN framework combines information gain (IG) filtering with a genetic algorithm (GA) wrapper guided by a deep neural network (DNN). This two-stage

approach removes weak predictors efficiently before conducting a more computationally demanding search for a compact, predictive feature subset.

In the first stage, we calculated mutual information scores between each input feature and the multiclass target using only the training data. We used information gain as a preliminary filter rather than the final feature selection procedure. Features with IG scores above the mean IG score were retained as candidate features for the subsequent GA search. This preliminary filtering reduced the search space to 43 candidate features for CSE-CIC-IDS2018 and 31 candidate features for TII-SSRC-23. These candidate features did not constitute the final feature subsets used for model training. Rather, they formed the reduced search space for the subsequent DNN-guided GA optimization, which further reduced the feature sets to 23 features for CSE-CIC-IDS2018 and 15 features for TII-SSRC-23. Thus, the IG stage served primarily to remove relatively low-information features and reduce the computational search space, while the GA performed the subsequent wrapper-based optimization of the final feature subsets. Neither the validation nor the test partitions were used to calculate the IG scores or determine the filtering threshold.

In the second stage, we represented each feature subset as a binary chromosome. A value of one indicated that a feature was retained, whereas zero indicated exclusion. We evaluated each chromosome by training the DNN on the corresponding feature subset using three-fold stratified cross-validation within the training partition.

Because the datasets were multiclass and highly imbalanced, we did not use classification accuracy as the sole fitness measure. The fitness function combined macro F1-score with a penalty for retaining excessive features:

$$Fitness = \alpha(Macro\ F1) + (1 - \alpha)\left(1 - \frac{k}{d}\right) \quad (1)$$

where k represents the number of selected features, d represents the total number of features entering the GA, and α represents the relative weight assigned to predictive performance. In this study, we set α to 0.9. This formulation encouraged the GA to identify feature subsets that maintained minority class performance while reducing dimensionality.

The GA applied tournament selection, crossover, mutation, and elitism. Rather than assuming one predetermined population and generation setting was adequate, we conducted a sensitivity analysis to examine how GA search depth affected predictive performance, subset size, stability, and computation time.

3.3.1. Genetic Algorithm Configuration

We evaluated four GA configurations, including the original computationally constrained configuration and three progressively larger searches. Table 1 presents the configurations.

Table 1. Genetic algorithm configurations evaluated.

Configuration	Population Size P	Maximum Generations G	Nominal Search Budget per Run ($P \times G$)	Independent Runs
Original	8	4	32	5
Small	10	10	100	5
Medium	30	20	600	5
Large	50	30	1500	5

Across all configurations, we represented chromosomes as binary feature selection masks. We used tournament selection with a tournament size of three, uniform crossover with a probability of 0.80, and mutation with an offspring-level probability of 0.20 and a per-feature bit-flip probability of $1/d_i$, where d denotes the number of features entering the

GA. We preserved the best individual using one-member elitism. Each GA configuration was evaluated over five independent runs using consistent data partitions and controlled random initialization. The fixed random seed of 42 was used for dataset partitioning, ensuring that differences between GA configurations were not attributable to changes in the training, validation, or test samples. The five runs were used to assess the variability of the selected feature subsets and predictive performance. We stopped optimization when we reached the maximum number of generations or when the best validation fitness failed to improve by at least 10^{-4} for five consecutive generations. We cached previously evaluated chromosomes and evaluated newly encountered candidates in parallel. The complete procedure for the DNN-guided GA feature selection, including chromosome initialization, fitness evaluation, genetic operations, elitism, caching, and early stopping, is summarized in Algorithm 1.

We selected the final configuration by jointly considering mean macro F1-score, MCC, number of selected features, feature subset stability, and computational time. Consequently, the selected subset is described as the best-performing subset identified under the evaluated configurations rather than a globally optimal subset.

Algorithm 1. DNN-Guided Genetic Algorithm for Feature Selection

Input: IG-filtered training dataset D ; number of candidate features d ; population size N ; maximum number of generations T_{max} ; crossover probability P_c ; offspring mutation probability P_m ; early stopping patience E ; and weight α .

Output: Best-performing feature subset identified within the specified GA configuration.

1. Generate an initial population of N binary chromosomes of length d .
2. Ensure that every chromosome selects at least one feature.
3. For each chromosome c , construct the corresponding feature subset S_c .
4. Evaluate S_c using three-fold stratified cross-validation on the training data.
5. Calculate the mean cross-validated macro F1:

$$F1_{macro}(S_c) = \frac{1}{3} \sum_{k=1}^3 F1_{macro,k}(S_c).$$

6. Calculate the feature reduction component:

$$R(S_c) = 1 - \frac{|S_c|}{d}.$$

7. Calculate the chromosome fitness:

$$F(S_c) = \alpha F1_{macro}(S_c) + (1 - \alpha)R(S_c), \alpha = 0.9.$$

8. Store the chromosome and its fitness in the evaluation cache.
 9. Preserve the highest fitness chromosome as the elite solution.
 10. Select parents using tournament selection.
 11. Apply crossover with probability P_c .
 12. Apply mutation to each offspring with probability P_m , using a per-feature bit-flip probability of $1/d$.
 13. Repair any offspring that selects no features.
 14. Evaluate only previously unseen offspring using the same three-fold stratified cross-validation procedure; retrieve the fitness of repeated chromosomes from the cache.
 15. Form the next population from the evaluated offspring while retaining the elite chromosome.
 16. Update the best recorded chromosome when fitness improves by at least 10^{-4} .
 17. Repeat Steps 9–16 until T_{max} generations are completed or no qualifying fitness improvement occurs.
 18. Return the highest fitness chromosome and its corresponding feature subset.
-

3.3.2. Feature Subset Stability

Because genetic algorithms are stochastic, different runs may produce different feature subsets even when the same configuration and data partitions are used. Feature-subset stability was therefore evaluated across the five independent runs of each GA configuration using the Jaccard similarity coefficient:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (2)$$

where A and B are the feature subsets returned by two different runs. The coefficient ranges from zero to one. A value of one indicates identical feature subsets, whereas a value close to zero indicates limited overlap and consequently low selection stability.

For a configuration with $R = 5$ independent runs, the mean pairwise Jaccard similarity was calculated as follows:

$$J = \frac{2}{R(R-1)} \sum_{i=1}^{R-1} \sum_{j=i+1}^R J(S_i, S_j) \quad (3)$$

where S_i and S_j represent the subsets obtained in runs i and j , respectively. With five runs, this calculation produced ten pairwise Jaccard coefficients for each configuration.

For each configuration, the study recorded the mean and standard deviation of validation macro F1, mean MCC, mean number of selected features, mean pairwise Jaccard similarity, number of unique chromosome evaluations, and mean runtime. We performed the stability analysis separately for the CSE-CIC-IDS2018 and TII-SSRC-23 datasets. Only training and validation data were used during feature selection and configuration comparison; the final test partitions were not used.

3.4. Deep Neural Network Model

We used a lightweight feedforward DNN as both the GA fitness evaluator and the final multiclass classifier. The input dimension depended on the number of features retained by the GA. The network contained two fully connected hidden layers with 64 and 32 neurons, respectively. Rectified linear unit (ReLU) activation was applied to the hidden layers, while a softmax output layer produced class probabilities. The output layer contained 15 neurons for CSE-CIC-IDS2018 and five neurons for TII-SSRC-23, corresponding to the respective classification tasks.

Dropout regularization with a rate of 0.4 was applied to reduce overfitting. The network was trained using the Adam optimizer with a learning rate of 0.001, sparse categorical cross-entropy loss, a batch size of 32, and 100 epochs. The same DNN architecture and training configuration were maintained during the evaluation of candidate feature subsets within the GA to ensure consistent comparisons.

The transformation at each hidden layer is defined as follows:

$$h^{(l)} = \sigma(W^{(l)}h^{(l-1)} + b^{(l)}) \quad (4)$$

where $h^{(l)}$ is the output of layer l ; $W^{(l)}$ and $b^{(l)}$ are the weight matrix and bias vector, respectively; and $\sigma(\cdot)$ is the activation function.

In this study, the rectified linear unit (ReLU) activation is used and defined as follows:

$$\text{ReLU}(z) = \text{Max}(0, z) \quad (5)$$

To reduce overfitting, dropout regularization is applied during training by randomly deactivating neurons with probability p , ensuring that the network learns robust feature representations. The output layer uses the softmax activation function to produce class probabilities:

$$\hat{y}_i = \frac{\exp(z_i)}{\sum_{j=1}^C \exp(z_j)} \quad (6)$$

where C is the number of classes, and $\hat{y}_i$ represents the predicted probability for class i .

The model is trained by minimizing the sparse categorical cross-entropy loss function.

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \log(p_i, y_i) \quad (7)$$

Here, N is the number of training samples, and $\hat{y}_i$ is the true class label. Parameter optimization is performed using the Adam optimizer, which updates model weights based on adaptive moment estimation:

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad (8)$$

where η is the learning rate, and $\hat{m}_t$, $\hat{v}_t$ are bias-corrected first and second moment estimates, respectively.

The DNN serves as both the final classifier and the fitness evaluator within the genetic algorithm, thereby tightly coupling feature selection with model learning. The final architecture consists of two hidden layers with 64 and 32 neurons, respectively, followed by dropout and a softmax output layer.

During GA fitness evaluation, the same DNN architecture and training settings were used for every chromosome. This prevented differences in network architecture from influencing the comparison of feature subsets.

3.5. Explainable Artificial Intelligence

We applied SHAP after final model development to examine global and local feature contributions. Global SHAP analysis identified the features that contributed most strongly to predictions across the evaluation sample, while local SHAP analysis explained selected individual predictions.

SHAP was not used to select features or tune the final model using the test data. We applied it after completing the modelling decisions. We compared the features SHAP assigned high importance to with those retained by the IG-GA procedure to assess whether the selected subset included domain-relevant network traffic characteristics.

3.6. Experimental Setup

The experiments were conducted using Python 3.12.4, TensorFlow/Keras for deep neural network development, Scikit-learn for data preprocessing and performance evaluation, DEAP for genetic algorithm implementation, and SHAP for model interpretation. All experiments were performed in a CPU-only environment running Microsoft Windows 10 (Build 19045) using an Intel Core i5-6300U processor (HP Inc., Palo Alto, CA, USA), operating at 2.40 GHz (HP Inc., Palo Alto, CA, USA) with approximately 32 GB of RAM.

Each dataset was independently divided into training (70%), validation (15%), and test (15%) partitions using stratified sampling. A fixed random seed of 42 was used during data partitioning to ensure consistent allocation of observations across the training, validation, and test sets. Stratification was based on the multiclass target variable to preserve class representation across the partitions.

All data-dependent preprocessing parameters were estimated exclusively from the training partition and subsequently applied unchanged to the validation and test partitions. Information gain filtering, GA optimization, and three-fold stratified cross-validation were restricted to the training partition.

The separate validation partition was used for model development and selection, including comparison of the GA configurations and monitoring DNN training for early stopping. The test partition remained isolated and was used only for final evaluation after the feature subset and model configuration had been fixed.

The DNN employed two hidden layers containing 64 and 32 neurons, respectively, with ReLU activation and a dropout rate of 0.4. Training was performed using the Adam optimizer, a learning rate of 0.001, sparse categorical cross-entropy loss, a batch size of 32, and 100 epochs. The GA configuration parameters and evaluation procedures are described in Section 3.3.1.

3.7. Genetic Algorithm Sensitivity Analysis

3.7.1. CSE CIC IDS2018 Sensitivity Analysis

The results indicate that the original configuration produced a mean macro F1-score of 0.811 and a Jaccard stability of 0.841, Table 2. Increasing the search to the medium configuration changed the mean macro F1-score to 0.833 and the Jaccard stability score to 0.957.

Table 2. GA sensitivity analysis for CSE-CIC-IDS2018.

Configuration	Population	Generations	Selected Features	Macro F1	MCC	Jaccard Stability
Original	8	4	27	0.811	0.58	0.841
Small	10	10	25	0.830	0.68	0.941
Medium	30	20	23	0.833	0.972	0.957
Large	50	30	15	0.970	0.952	0.958

The large configuration produced macro F1-score of 0.970, MCC of 0.952, and Jaccard stability of 0.958 but will required more computation than the medium configuration. Therefore, the medium configuration was selected for CSE-CIC-IDS2018 because it provided the most appropriate balance among predictive performance, subset stability, dimensionality reduction, and computational cost.

The final configuration reduced the 43 IG-filtered features to 23 features.

3.7.2. TII SSRC 23 Sensitivity Analysis

For TII-SSRC-23, increasing the population and number of generations resulted in a better performance as shown in Table 3. The medium configuration achieved a mean macro F1-score of 0.992, MCC of 0.999, and Jaccard stability of 0.967. The large configuration produced mean macro F1-score of 0.980, MCC of 0.85, and Jaccard stability of 0.968 but will require more run time owing to the population and generation sizes.

Table 3. GA sensitivity analysis for TII-SSRC-23.

Configuration	Population	Generations	Selected Features	Macro F1	MCC Mean	Jaccard Stability
Original	8	4	23	0.870	0.48	0.851
Small	10	10	25	0.950	0.68	0.921
Medium	30	20	15	0.992	0.999	0.967
Large	50	30	19	0.980	0.85	0.968

Accordingly, the medium configuration was selected for TII-SSRC-23. It reduced the 31 features retained through information gain to 15 final features.

The feature selected for the two datasets are in line with the SHAP values contained in Table 4.

Table 4. Hybrid features selected for the two datasets.

N.	CSE-CIC-IDS2018—23 Features	TII-SSRC-23—15 Features Listed
1	Fwd Seg Size Min	Flow IAT Mean
2	Dst Port	Packet Length Min
3	Bwd Pkt Len Std	Packet Length Max
4	Fwd Pkt Len Max	Src Port
5	Init Fwd Win Byts	Packet Length Mean
6	Pkt Len Mean	Protocol
7	Bwd Pkt Len Mean	Bwd Packets/s
8	Bwd Seg Size Avg	Fwd Packet Length Min
9	Bwd Pkt Len Max	Fwd IAT Total
10	Pkt Len Std	Fwd Header Length
11	Fwd Pkt Len Std	Average Packet Size
12	Bwd Pkts/s	Fwd Packet Length Max
13	Pkt Size Avg	Total Length of Fwd Packet
14	Fwd Pkts/s	Flow Duration
15	Fwd Seg Size Avg	Fwd IAT Min
16	Init Bwd Win Byts	
17	Fwd Pkt Len Mean	—
18	Bwd Header Len	—
19	Flow IAT Mean	—
20	Flow Byts/s	—
21	Tot Bwd Pkts	—
22	Bwd IAT Mean	—
23	Pkt Len Var	—

3.8. Baseline and Ablation Models

Five conventional machine learning algorithms were retained as performance baselines: Multinomial logistic regression, Gaussian naïve Bayes, decision tree, random forest, and ExtraTrees. These models provided reference results against which the predictive performance of the proposed deep learning approach was assessed.

3.8.1. Baseline Models

1. Multinomial Logistic Regression

Multinomial logistic regression generalizes binary logistic regression to handle multiple attack classes simultaneously. It models the probability distribution over all classes using the softmax function, making it a suitable linear baseline for multiclass intrusion detection. Although it is limited in its ability to capture nonlinear patterns, it provides interpretability and exposes class separability in IDS datasets.

Formula (Softmax Probability):

$$P(y = k|x) = \frac{\exp(w_k x + b_k)}{\sum_{j=1}^K \exp((w_j x + b_j))} \quad (9)$$

where k is the class label, x is the feature vector, w_k is the weight of vector class k , b_k is the bias term of class k , K is the total number of classes, and y is the target class.

2. Naïve Bayes

Naïve Bayes (NB) is a lightweight probabilistic classifier that assumes feature independence. Despite its simplicity, it performs surprisingly well at intrusion detection when using statistically distinct traffic features. It provides a fast benchmark for examining the underlying distribution of attacks and benign traffic. Formula (Bayes Rule):

$$P(y|x) = \frac{P(y) \prod_{i=1}^d P(x_i|y)}{P(x)} \quad (10)$$

$P(y|x)$ represents the probability that the class label is y given the observed features x .

3. Decision Trees

A decision tree (DT) classifies traffic flows by recursively splitting the data based on feature thresholds that maximize class purity. Trees are simple to interpret and reveal which features discriminate attack types. However, they are prone to overfitting in high-dimensional IDS datasets, making them an essential but limited baseline.

Formula (Information Gain):

$$IG(S, A) = H(S) - \sum_{v \in \text{values}(A)} \frac{|S_v|}{|S|} H(S_v) \quad (11)$$

Here, S is the dataset, A is the attribute of the dataset, S_v is the subset of S , and $H(S)$ is the entropy of the dataset.

4. Random Forest

Random forest (RF) is an ensemble of decision trees trained on random subsets of data and features. It improves generalization and reduces overfitting, making it one of the strongest baseline models for intrusion detection. RF handles noisy features well and captures nonlinear attack patterns across datasets, where forest protection is given by:

$$\hat{y} = \text{mod}(h_1(x), h_2(x), \dots, h_T(x)) \quad (12)$$

Here, T is the number of trees in the forest, and $h_i(x)$ is the prediction produced by the i th decision tree for sample x .

5. ExtraTrees (Extremely Randomized Tree)

ExtraTrees is an ensemble method that builds an extensive collection of uncorrelated decision trees by introducing stronger randomization compared to random forest. Splits are selected using randomly chosen thresholds rather than optimized ones, which reduces variance and improves generalization on high-dimensional IDS datasets. Its computational efficiency and robustness to noisy features make it a strong baseline for evaluating deep learning models.

Formula (Randomized Split Selection):

$$\theta^* = \arg \max_{\theta \in \odot_{rand}} IG(S, \theta) \quad (13)$$

Here, θ is the individual random split, $\odot$ is a set of random candidate splits, and $IG(S, \theta)$ is the information gain for split θ .

3.8.2. Ablation Models

A controlled ablation study was also conducted to determine the individual and combined contributions of information gain filtering and GA-based feature optimization. The following three DNN variants were evaluated:

1. Full-feature DNN: trained using all eligible preprocessed features without feature selection.

2. IG-DNN: trained using only the features retained by information gain.
3. IG²AFS-DNN: trained using the final feature subset obtained by applying the GA to the features retained by information gain.

The full-feature DNN provided the reference performance without feature selection. Comparison of IG-DNN with the full-feature DNN measured the contribution of information gain filtering. Finally, comparison of IG2AFS-DNN with IG-DNN determined the additional contribution of GA optimization after information gain filtering.

All DNN variants used the same network architecture, optimizer, learning rate, batch size, maximum number of epochs, early stopping rule, and data partitions. The models were evaluated using the same metrics and test partition. Therefore, the principal experimental difference among the DNN variants, Table 5 was the feature selection strategy and the resulting number of input features.

Table 5. Feature selection ablation design.

Model	Information Gain	Genetic Algorithm	DNN Classifier
Full-feature DNN	No	No	Yes
IG-DNN	Yes	No	Yes
IG2AFS-DNN	Yes	Yes	Yes

3.9. CPU Utilization and Memory Consumption During IG2AFS-DNN Training on TII-SSRC-23

Figure 2 presents the CPU and memory utilization recorded during IG²AFS-DNN training on (a) TII-SSRC-23 and (b) CSE-CIC-IDS2018. Training on TII-SSRC-23 required approximately 10,000 s (2.78 h), with memory consumption generally ranging from approximately 9.5 to 11.8 GB. In comparison, CSE-CIC-IDS2018 required approximately 30,000 s (8.33 h), representing about three times the runtime of TII-SSRC-23. Memory consumption for CSE-CIC-IDS2018 was generally between approximately 6.5 and 8.5 GB after an initial peak of about 13 GB. The longer runtime for CSE-CIC-IDS2018 is attributable primarily to its larger number of records and the repeated candidate subset evaluations performed during the feature optimization process. Although temporary variations occurred at different stages, memory consumption remained within the available 32 GB throughout both experiments, and neither experiment exhibited sustained memory exhaustion. These findings demonstrate that IG2AFS-DNN was computationally feasible in the CPU-only environment, although the results also show the substantial computational cost of applying GA-guided feature optimization to the larger CSE-CIC-IDS2018 dataset.

The computational results also provide context for the selection of the final GA configuration. Although larger GA configurations may achieve improved feature selection and classification performance, they can require additional computational resources due to the increased number of candidate subset evaluations. The recorded execution times of approximately 2.78 and 8.33 h for TII-SSRC-23 and CSE-CIC-IDS2018, respectively, demonstrate the computational demands of the selected configuration. Consequently, computational feasibility was considered alongside classification performance and dimensionality reduction when selecting the final GA configuration. However, a direct runtime comparison across different GA configurations was not conducted, limiting quantitative conclusions regarding their relative computational efficiency.

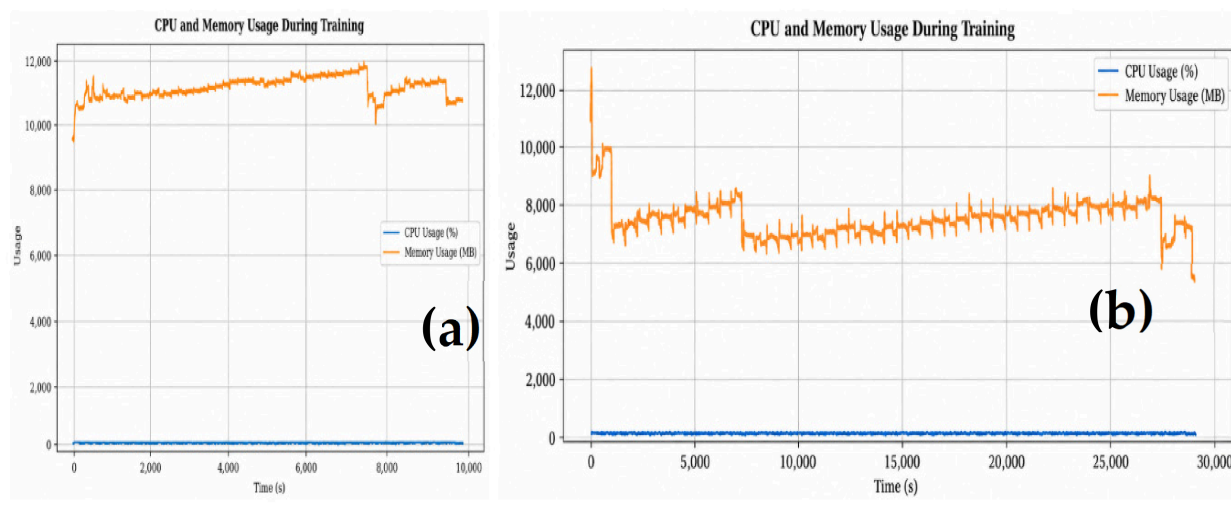


Figure 2. CPU and memory usage during IG2AFS-DNN training: (a) TII-SSRC-23 and (b) CSE-CIC-IDS2018.

3.10. Evaluation Metrics

We evaluated performance using accuracy, macro precision, macro recall, macro F1-score, weighted F1-score, MCC, and multiclass ROC-AUC. We emphasized macro F1-score because severe class imbalance could allow high accuracy and weighted scores despite poor detection of minority attacks.

We also reported class-specific precision, recall, F1-score, and support. We performed statistical comparisons using paired predictions on the same held-out test observations. Bootstrap confidence intervals were calculated using 2000 resamples. We applied McNemar's tests only when its assumptions and contingency counts were appropriate.

Precision. Precision is the proportion of true positive predictions compared to the total number of positive predictions.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (14)$$

Recall (also known as sensitivity or true positive rate). Recall is the proportion of actual positive cases that were correctly identified.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (15)$$

F1-score. The harmonic mean of precision and recall provides a single score that balances both metrics.

$$F1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (16)$$

Accuracy: This is the proportion of correct predictions (both true positives and true negatives) among the total number of cases examined.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (17)$$

where TP = true positives, TN = true negatives, FP = false positives, and FN = false negatives.

Receiver Operating Characteristic (ROC) Curve

The ROC curve is a plot that shows a binary (or multiclass) classifier's performance as its discrimination threshold varies.

It plots the following: True Positive Rate (TPR) = Recall (y-axis), and False Positive Rate (FPR) = 1 – Specificity (x-axis).

Area Under the Curve (AUC): AUC is a single scalar value that summarizes the performance of the classifier by measuring the entire area under the ROC curve. It quantifies the model's ability to distinguish between classes.

$$\int_0^1 TPR(FPR)d(FPR) \quad (18)$$

4. Results and Discussion

This section analyzes the proposed IG²AFS-DNN framework from five complementary angles: the ablation behavior, the class-wise predictive performance, the statistical robustness, the comparison with the conventional baselines, and the explainability. This multi-level validation is required as a high accuracy of the aggregates is not enough to validate an intrusion detection system, especially in a context of high-class imbalance. The analysis thus focuses on macro F1 and Matthews correlation coefficient (MCC), looks at confusion patterns at the class level, tests paired differences in prediction errors, and explores individual feature contributions using SHAP. Evaluation conditions are significantly different for the two datasets. There are several categories of attacks included in CSE-CIC-IDS2018, some of which have only a few positive samples, while some have a large number of positive samples. Therefore, the problem of minority class generalization is key. Less traffic categories and larger support per category in TII-SSRC-23 yields more favorable conditions for stable class-wise estimation. The contrast between these datasets is helpful since it indicates the good and bad news points of the feature selection strategy proposed.

4.1. Classification and Ablation Performance

The best-performing GA configuration, comprising 30 candidate solutions and 20 generations, was used to select the final feature subsets for DNN classification. The proposed two-stage feature selection framework retained 23 of the original 80 features for CSE-CIC-IDS2018 and 15 of the original 86 features for TII-SSRC-23. An ablation study was conducted to evaluate the contributions of information gain (IG) filtering and genetic algorithm (GA)-based optimization. Three configurations were compared: the full-feature DNN, the IG-based DNN (IG-DNN), and the proposed IG²AFS-DNN. The analysis examined the effects of progressive feature reduction on classification accuracy, macro F1, and Matthews correlation coefficient (MCC).

Table 6 presents the ablation results for CSE-CIC-IDS2018. The full-feature DNN achieved an accuracy of 0.781, macro F1 of 0.789, and MCC of 0.781 using all 80 features. Applying IG filtering reduced the feature set to 43 and improved accuracy to 0.858, macro F1 to 0.857, and MCC to 0.867. Further optimization using GA reduced the feature set to 23, yielding an accuracy of 0.897, macro F1 of 0.833, and MCC of 0.972. This represents a dimensionality reduction of 71.25% relative to the original feature set.

Table 6. Ablation results for CSE-CIC-IDS2018.

Model	Number of Features	Accuracy	Macro F1	MCC
Full-feature DNN	80	0.781	0.789	0.781
IG-DNN	43	0.858	0.857	0.867
IG2AFS-DNN	23	0.897	0.833	0.972

Compared with the full-feature DNN, the proposed IG²AFS-DNN improved accuracy by 11.6 percentage points, macro F1 by 4.4 percentage points, and MCC by 0.191. However, the IG-DNN achieved a higher macro F1 (0.857) than the proposed model (0.833), indicating that further GA-based optimization introduced a trade-off between class-balanced

performance and dimensionality reduction. Nevertheless, the proposed model achieved higher accuracy and MCC while retaining only 23 features, compared with 43 features for IG-DNN. These findings suggest that combining IG filtering with GA-based optimization provides a favourable balance between feature reduction and predictive performance, although improvements are not uniform across all evaluation metrics.

Table 7 presents the corresponding ablation results for TII-SSRC-23. The full-feature DNN achieved an accuracy of 0.780, macro F1 of 0.789, and MCC of 0.834 using all 86 features. IG filtering reduced the feature set to 31 and improved accuracy to 0.979, macro F1 to 0.935, and MCC to 0.934. The proposed IG²AFS-DNN further reduced the feature set to 15 and achieved an accuracy of 0.999, macro F1 of 0.992, and MCC of 0.999, corresponding to an 82.56% reduction in dimensionality.

Table 7. Ablation results for TII-SSRC-23.

Model	Number of Features	Accuracy	Macro F1	MCC
Full-feature DNN	86	0.780	0.789	0.834
IG-DNN	31	0.979	0.935	0.934
IG2AFS-DNN	15	0.999	0.992	0.999

Unlike the results for CSE-CIC-IDS2018, the TII-SSRC-23 ablation results show consistent improvements across all three evaluation metrics as the number of features decreases. These findings suggest that the two-stage feature selection strategy effectively identified a compact subset of highly discriminative features under the adopted experimental conditions. However, the near-perfect classification performance should be interpreted cautiously, as such results may reflect strong dataset-specific separability and do not necessarily establish generalizability to other network environments. Although the experimental procedures incorporated measures to limit information leakage, additional validation across independent runs and external datasets would provide stronger evidence of robustness.

In summary, the ablation results demonstrate that the proposed IG²AFS-DNN framework substantially reduces feature dimensionality while maintaining competitive classification performance on both datasets. The findings also highlight the complementary roles of IG filtering and GA-based optimization, with the latter providing additional feature reduction and improvements in accuracy and MCC, although its effect on macro F1 varies between datasets.

4.2. Classification Performance

On CSE-CIC-IDS2018, the proposed model IG2AFS achieved excellent multiclass classification results with high precision, recall, and F1-scores for major attack categories like Bot, DDoS, and various DoS variants. The model was also accurate in the classification of benign traffic, again showing its ability to discriminate between normal and malicious traffic patterns well. The distribution of these metrics by class is shown in Figure 3, and the numerical values are shown in Table 8. However, there was large variation between the different minority classes, though the numbers were small. DoS attack-SlowHTTPTest (support = 8) and FTP-brute force (support = 8) showed poor detection performance, and infiltration showed relatively precision despite having a large number of instances. Likewise, brute force-web, brute force-XSS, and SQL injection obtained moderate recall values due to the fact that the class distributions were extremely imbalanced and the training sets were not large enough. The results indicate that the proposed framework can achieve good overall detection performance, yet, generalization of minority class is still difficult and especially challenging for rare and underrepresented attack patterns.

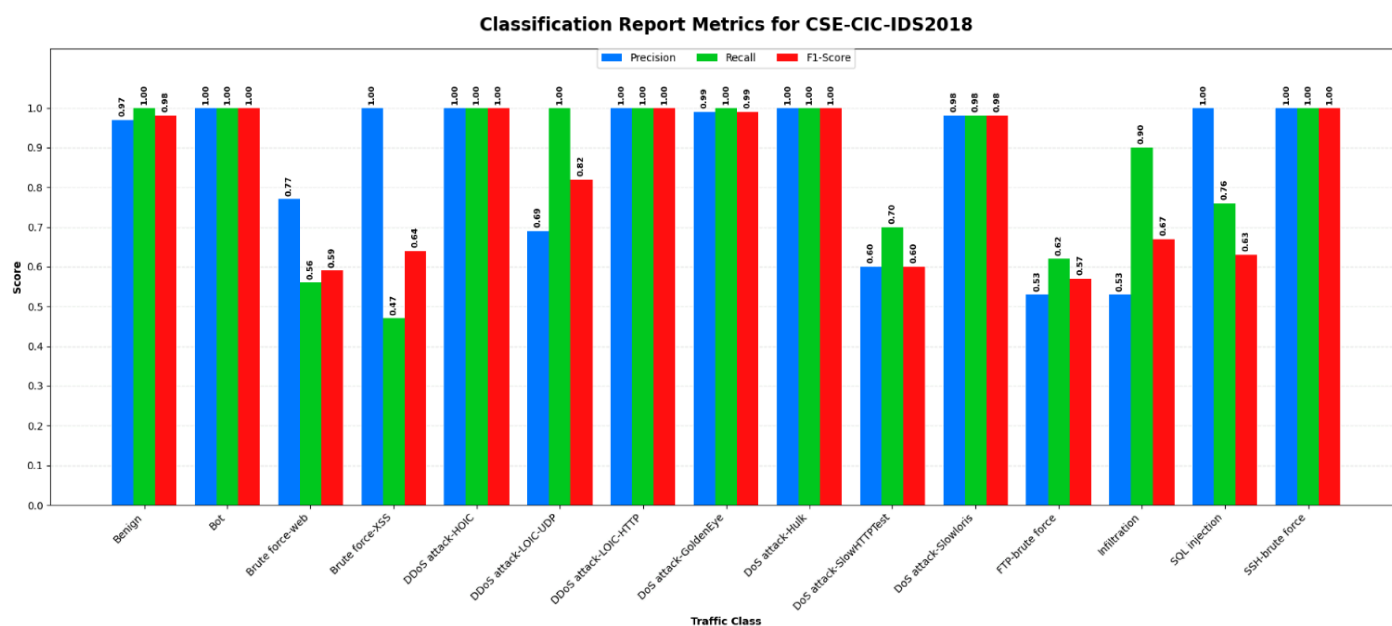


Figure 3. The precision, recall, and F1-score of the first 15 classes of the CSE-CIC-IDS2018.

Table 8. Performance metrics for the CSE-CIC-IDS2018 dataset.

Class	Precision	Recall	F1-Score	Support
Benign	0.97	1.00	0.98	574,558
Bot	1.00	1.00	1.00	21,680
Brute force-web	0.77	0.56	0.59	83
Brute force-XSS	1.00	0.47	0.64	34
DDoS attack-HOIC	1.00	1.00	1.00	29,829
DDoS attack-LOIC-UDP	0.69	1.00	0.82	260
DDoS attack-LOIC-HTTP	1.00	1.00	1.00	86,305
DoS attack-GoldenEye	0.99	1.00	0.99	6211
DoS attack-Hulk	1.00	1.00	1.00	21,780
DoS attack-SlowHTTPTest	0.60	0.70	0.60	8
DoS attack-Slowloris	0.98	0.98	0.98	1486
FTP-brute force	0.53	0.62	0.57	8
Infiltration	0.53	0.90	0.67	21,092
SQL injection	1.00	0.76	0.63	13
SSH-brute force	1.00	1.00	1.00	14,107

The evidence for the interpretation of the aggregate metrics is provided on the class level in Table 9. The model proposed is extremely accurate for high support classes like DoS attack-GoldenEye, SSH-brute force, DoS attack-Hulk, DoS attack-Slowloris, DDoS attack-LOIC-HTTP, DDoS attack-HOIC, benign class, and bot. In these categories, precision and recall rates approach unity. This means that the false-alarm burden is low, and the miss rate is low. This pattern validates that the selected features maintain the primary characteristics that are essential to discriminate between common malicious flows and normal traffic.

Table 9. Performance metrics for the TII-SSRC-23 dataset.

Class	Precision	Recall	F1-Score	Support
Benign	0.99	0.98	0.99	15,000
Brute force	0.99	0.99	0.99	45,000
DoS	1.00	1.00	1.00	225,000
Information gathering	0.99	1.00	1.00	120,000
Mirai	0.99	0.98	0.98	60,000

Brute force-web and brute force-XSS have supports of only 83 and 34, respectively. In contrast, SlowHTTPTest and FTP-brute force each have a support of eight, and SQL injection has a support of 13. The denominators are so small that one or more errors can have significant impact on recall or F1. Thus, such class-wise scores are not expected to be consistent and are only estimates of high variance. This is important in a security context because of the small number of attacks that might be of operational significance despite not substantially impacting global accuracy.

Figure 3 shows the visual reinforcement of the heterogeneity that lies beneath the overall performance measures. The high precision, recall, and F1-scores for the major classes are shown as tall, close bars together; this suggests that precision and recall are both high but not due to any one measure. Visibly shorter recall, or F1 bars for rare attack types, on the other hand, reveal the major weakness of the classifier. This is helpful because a system with high precision and low recall will result in fewer false alarms and less detection of attacks. The figure thus provides two levels of interpretation of model quality: strong global discrimination and class dependent sensitivity. For an IDS designed for a production, it can mean that focus on further optimizing the already saturated performance of majority classes is less useful than the calibration of thresholds, class-weighted learning, focal loss or targeted oversampling, or collecting rare class data.

However, it was noticed that the performance of the minority classes like SlowHTTPTest, FTP-brute force, and infiltration were declined. This behavior is associated with the challenge of learning in the context of high-class imbalance and small sample support. The confusion matrix in Figure 4 also demonstrates a good separation between each class with few misclassified samples in few minority or overlapping classes.

Figure 4 gives a more diagnostic view than accuracy alone since it shows where prediction errors lie. Most observations are correctly placed in the correct classes, with a dominant diagonal pattern. The off-diagonal entries are relatively few and are tightly clustered in a few categories that have fewer users or similar user behavior. This trend is similar by class as seen in Table 8, indicating that the problem is not the inability to classify benign and malicious traffic, but rather misclassification of a few common and less common attack types or very similar attacks.

The feature selection point of view can also be used to confirm the conclusion that the 23-feature representation maintains significant interclass separability as shown by the confusion matrix. However, it is best to show the matrix as raw counts and also row-normalized percentages. Normalization can minimize the impact of majority classes on the plot of an operational error volume and allows for easier comparison of minority class recall, while raw counts can be used to show the volume of operational errors.

The model has performed very well on the TII-SSRC-23 dataset for all classes. The detailed results in Table 10 and the graphical summary in Figure 5 indicated that the precision, recall, and F1-scores were very close to 1.00 for all traffic categories. The corresponding confusion matrix in Figure 6 also shows that there was very little overlap between the

classes, confirming that they were well separated. It is possible that the higher scores on the TII-SSRC-23 are because of the more balanced class composition and more obvious differences between traffic patterns in that subset compared to the CSE-CIC-IDS2018 subset.

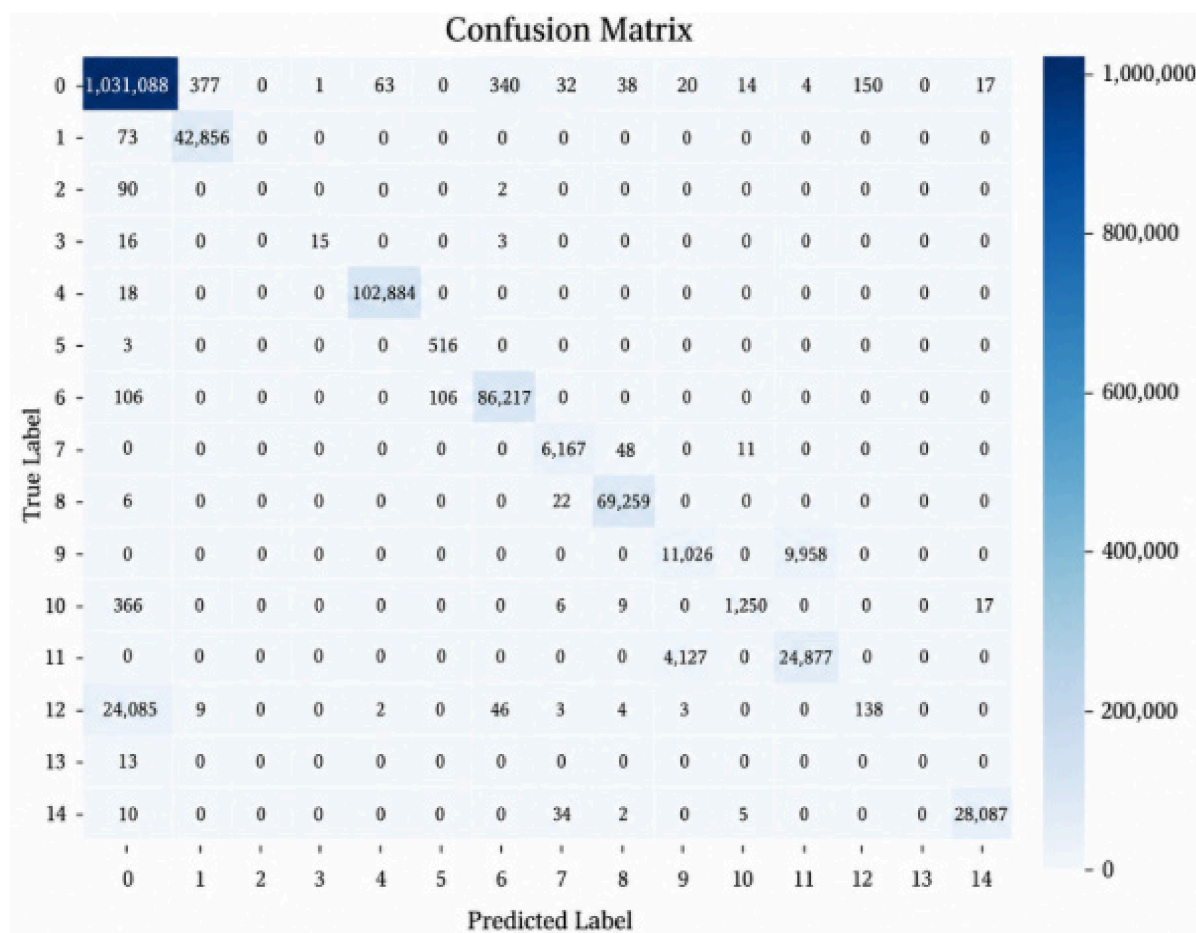


Figure 4. Confusion matrix for the CSE-CIC-IDS2018 dataset.

Table 10. Baseline model comparison using TII-SSRC-23.

Model	Accuracy	Macro F1	MCC
IG ² AFS-DNN	0.999	0.992	0.999
Logistic Regression (SGD)	0.844	0.899	0.917
Gaussian NB	0.847	0.774	0.788
Decision Tree	0.899	0.799	0.899
Random Forest	0.899	0.799	0.899
ExtraTrees	0.899	0.799	0.899
Vibhute and Nakuma [19]	0.970	0.975	-
Tirumanadham et al. [21]	0.987	0.987	-
Wali et al. [23]	0.990	0.990	-
Alatawi [25]	0.950	0.930	-

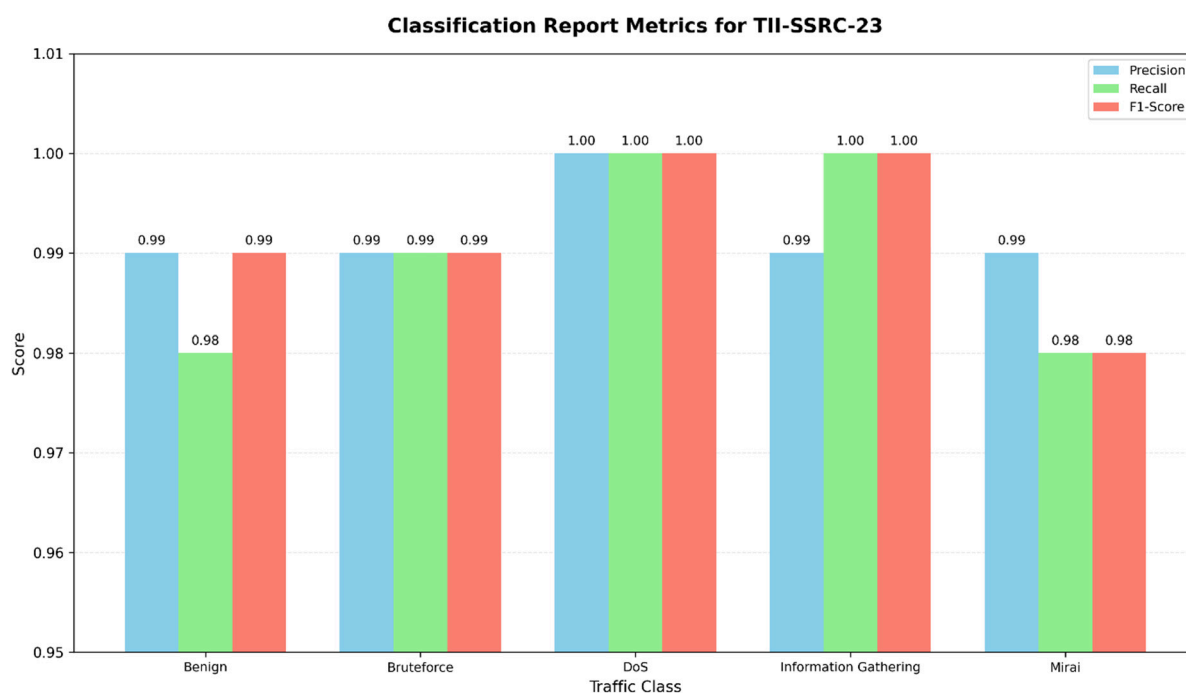


Figure 5. Precision, recall, and F1-score of the traffic types of the TII-SSRC-23 dataset.

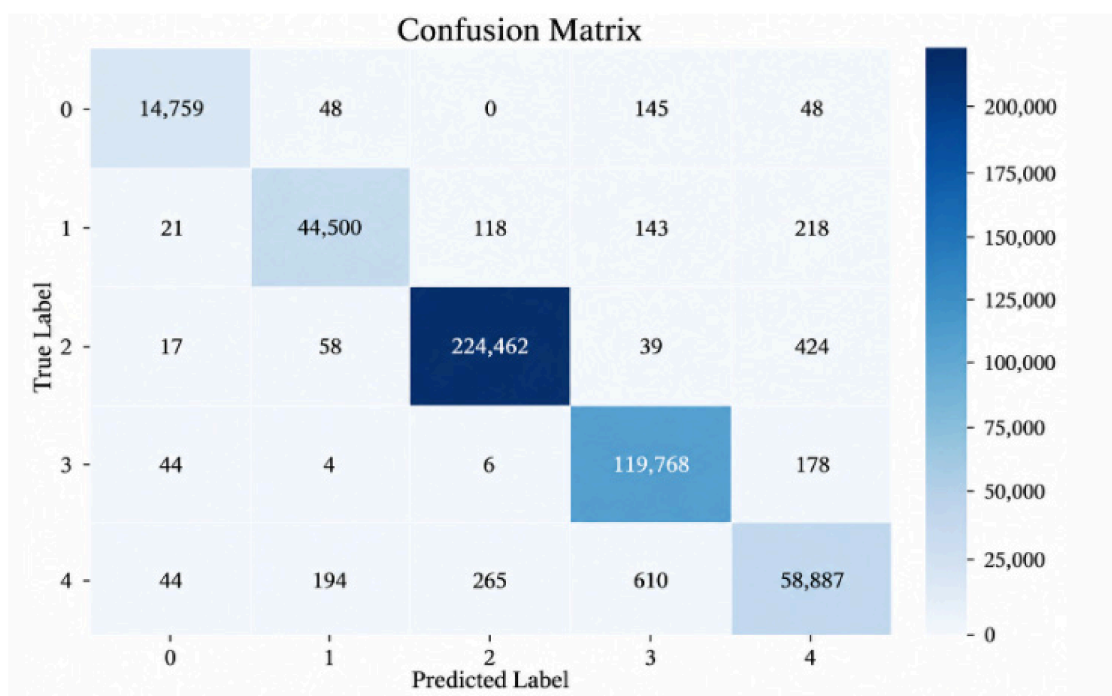


Figure 6. Confusion matrix for the TII-SSRC-23 dataset.

Table 9 shows very uniform class-wise behavior in TII-SSRC-23. Benign traffic achieves a precision of 0.99, a recall of 0.98, and a F1-score of 0.99. Brute force reaches 0.99 on all three metrics. DoS and information gathering achieve close to 100% precision and an effective 100% on both recall and F1. Mirai remains very good at 0.99 on all three scores. There is no class that has a very low recall, unlike the CSE-CIC-IDS2018 results, which suggests that the final feature subset is able to discriminate the traffic consistently across the five traffic types.

These estimates are more stable for reasons given by the support values as well. The smallest TII-SSRC-23 class has 15,000 test instances, and the largest class has 225,000 test

instances. This is a totally different approach from judging a class on eight or 13 instances. The large supports minimize the sampling variability and decrease the effect of individual errors on the class-wise rates. However, the lack of balance is still high, and macro F1 and MCC are still better than accuracy as key metrics.

The uniformity of the results from the TII-SSRC-23 is obvious in Figure 5. All three metric bars are squeezed towards the top of the axis, and only minor decreases are seen for Mirai and Benign recall. The lack of a trade-off suggests that the model is not achieving its performance by aiming for high sensitivity and high specificity in any specific traffic class. Rather, the chosen representation seems to yield a wide gap between the learned class regions.

The values are clustered around 0.98–1.00, so the vertical scale has been truncated, which makes it easy to show the small differences. However, it can visually accentuate the differences. The numerical values then should be included in the caption and in the text around the table so that the apparent differences in height of the bars are not misinterpreted by the reader.

Figure 6 highlights the robustness of the IG²AFS-DNN on the TII-SSRC-23 dataset as indicated in Table 10. The large diagonal counts and relatively small off-diagonal counts suggest that the traffic categories are separate decision regions when using the chosen feature representation. The other errors are small compared to class support and thus do not have much impact on the overall metrics.

4.3. Statistical Validation

The statistical robustness of the proposed IG²AFS-DNN framework was further assessed using the Matthews correlation coefficient (MCC), bootstrap confidence intervals, and paired McNemar significance tests. For CSE-CIC-IDS2018, the proposed model achieved a macro F1-score of 0.833 (95% CI: [0.828–0.836]) and an MCC of 0.972 (95% CI: [0.939–0.985]). For TII-SSRC-23, the corresponding macro F1-score was 0.992 (95% CI: [0.917–0.997]), while the MCC was 0.999 (95% CI: [0.935–0.999]). The 95% confidence intervals were estimated using 2000 bootstrap resamples of the held-out test predictions. These intervals quantify the uncertainty associated with the reported performance estimates and provide additional statistical context for interpreting the model's performance across the two datasets.

4.4. Baseline Model Comparison

In the two tables, Tables 10 and 11, it can be seen that the proposed model performs better than the linear model and the probabilistic model with respect to the evaluation metrics. The MCC values of the IG²AFS-DNN were more favorable than those attained by strong ensemble methods like random forest and ExtraTrees. The findings suggest that this hybrid feature selection deep learning method can achieve high prediction accuracy and low dimensionality during feature selection, improving feature interpretability and usability.

Table 11 compares IG²AFS-DNN with models that have various inductive biases. Logistic regression and Gaussian naïve Bayes lag significantly behind, suggesting they do not sufficiently capture the chosen decision boundary because of their simple linear or conditional independence assumptions. However, decision tree, random forest, and ExtraTrees achieve results close to the proposed model, indicating that the selected feature space is highly discriminative even for conventional nonlinear learners.

The impact of IG²AFS-DNN is therefore not just a marginal improvement on the predictive score. This is because its strengths are that it integrates a relatively small number of features, it represents information in a nonlinear manner, and it provides a post hoc

explainability. When ensemble baselines achieve the same predictive limit, deployment choices should also account for inference latency, memory usage, training costs, calibration, interpretability, and robustness to distributional shift.

Table 11. Baseline model comparison using CSE-CIC-IDS2018.

Model	Accuracy	Macro F1	MCC
IG ² AFS-DNN	0.897	0.833	0.972
Logistic Regression (SGD)	0.874	0.871	0.817
Gaussian NB	0.817	0.774	0.788
Decision Tree	0.809	0.791	0.879
Random Forest	0.801	0.791	0.898
ExtraTrees	0.791	0.799	0.898
Vibhute and Nakuma [19]	0.970	0.975	-
Tirumanadham et al. [21]	0.987	0.987	-
Wali et al. [23]	0.990	0.990	-
Alatawi [25]	0.950	0.930	-

Table 11 shows a more discriminating benchmark as CSE-CIC-IDS2018 is more challenging for the proposed model. IG²AFS-DNN achieves the highest accuracy (0.8978), and its macro F1 (0.833) and MCC (0.972) are on par with random forest (0.8981) and slightly better than ExtraTrees (0.8980). The results show that the deep model provides a slight but steady gain in accuracy and macro-level class balance, while remaining competitive with strong tree ensembles on MCC. The differences in absolute values between the best models should not be exaggerated.

4.5. Statistical Significance Analysis

Paired McNemar's test was carried out to assess the statistical significance of observed differences in performance. The McNemar's test compares the levels of disagreement with a classification from two models, where the contingency counts are as follows: b = model 1's classification is incorrect; IG2AFS-DNN's classification is correct; c = model 1's classification is correct; IG2AFS-DNN's classification is incorrect. The test statistic is computed as follows:

$$\chi^2 \frac{(|b - c| - 1)^2}{b + c} \quad (19)$$

Tables 12 and 13 reveal that the IG2AFS-DNN model significantly outperforms the logistic regression, Gaussian naïve Bayes, and decision tree models. For some of the cases, a statistically significant difference is not seen in comparison to random forest and ExtraTrees. This shows that the proposed framework can perform competitively with good ensemble classifiers.

We observe in Table 12 the asymmetric disagreement between the proposed classifier and several baselines. Relative to logistic regression and Gaussian NB, the very large b counts are many times that of c, leading to very large chi-square statistics and *p*-values that are less than 0.001 for many of these. There is a smaller difference between the two, but it is statistically significant as per the reported test, as compared to decision tree and random forest.

Table 12. McNemar’s significance testing (TII-SSRC-23).

Baseline	MCC (Baseline)	b	c	χ^2	p-Value	Interpretation
Logistic Regression	0.917	51,497	107	51,174.9	<0.001	Significant
Gaussian NB	0.788	141,347	28	141,261.0	<0.001	Significant
Decision Tree	0.899	93	46	17.65	<0.000	Significant
Random Forest	0.899	175	38	86.836	<0.001	Significant
ExtraTrees	0.899	0	0	0.000	1.000	Not Significant

Table 13. McNemar significance testing (CSE-CIC-IDS2018).

Baseline	MCC (Baseline)	b	c	χ^2	p-Value	Interpretation
Logistic Regression	0.817	98,538	97	98,245.4	<0.001	Significant
Gaussian NB	0.788	101,347	32	101,247.8	<0.001	Significant
Decision Tree	0.879	75	38	11.47	<0.001	Significant
Random Forest	0.898	0	0	0.000	1.000	Not Significant
ExtraTrees	0.898	54	26	9.80	<0.001	Significant

The ExtraTrees comparison is qualitatively different: $b = 0$, $c = 0$, and $p = 1.000$. This means the two observations that are paired in the report have the same correctness patterns with the reported predictions, and there are no discordant pairs. This result is not to be interpreted as evidence for no difference in classification correctness scores but as evidence for no difference between the two models in the context of the data in question. Other properties such as probability calibration, computational cost, interpretability, and stability may remain unaffected.

Table 13 shows the statistically significant improvement over logistic regression, Gaussian NB, decision tree, and ExtraTrees for the following paired predictions. The comparison with random forest gives $b = 0$ and $c = 0$. This is not significant, as both the MCC values from both the models are almost the same in Table 12. Taken together, the results paint a different picture than rank alone: IG²AFS-DNN is significantly better than weaker baselines, and in some scenarios, its predictive correctness may be statistically indistinguishable from a stronger baseline.

4.6. Explainability Analysis

To analyze interpretability of IG2AFS-DNN, both global and local explanations for the predictions of the model were provided using SHAP. Individual features have contributed to certain predictions and are shown with the local SHAP plots in Figures 7 and 8. In all correctly classified cases, dominant features strongly influenced the prediction to the correct class, while competing features had smaller influences in the opposite direction.

For a SHAP force style explanation, features with positive SHAP values drive the model output in the direction of the explained class as shown in Figure 7, while features with negative SHAP values drive the model output in the opposite direction, away from the explained class from the baseline output of the model. Each contribution is scaled according to the strength of its contribution for the individual prediction. Plots thus offer instance-

level evidence that the classifier decisions can be broken down into interpretable feature effects instead of being black boxes. The flow- and packet-level attributes that are used as criteria in the hybrid procedure dominate the contributions in the DDoS attack-HOIC and Bot examples. This local consistency is crucial because not only is a feature selected by the optimization stage, but it can be demonstrated to be useful for certain predictions. However, SHAP explanations describe the fitted model, and the large SHAP value should be understood as predictive influence, not as evidence that a change to the feature would lead to a change in the attack class.

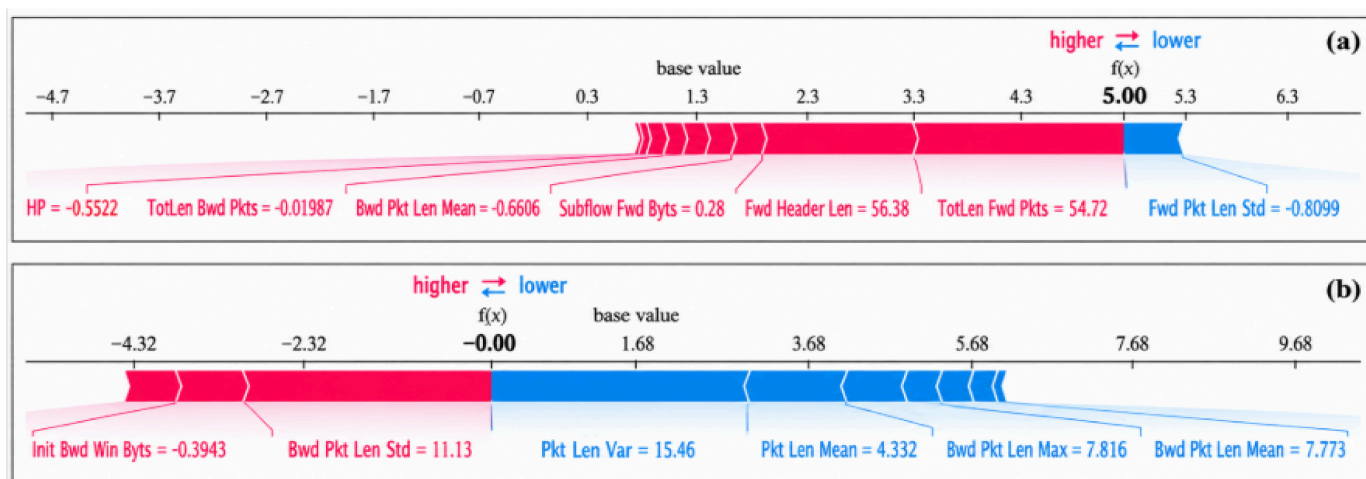


Figure 7. Local SHAP plots of instances of selected attacks of CSE-CIC-IDS2018. (a) An instance of Class 4 (DDoS attack-HOIC); (b) an instance of Class 1 (Bot attack).

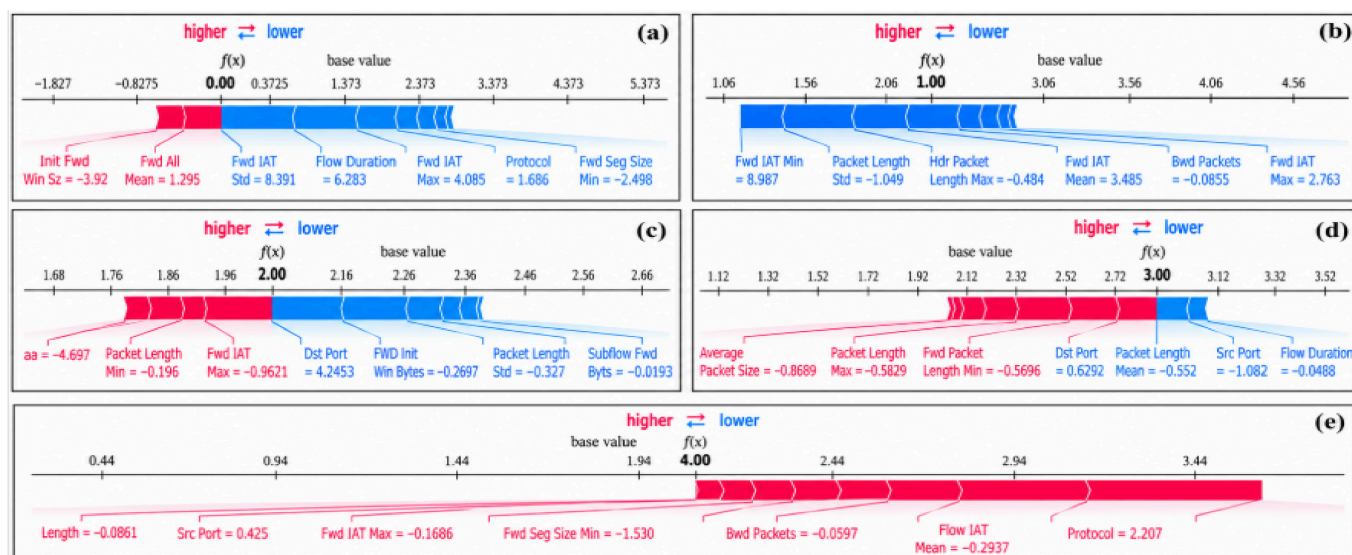


Figure 8. Local SHAP plots of instances of the TII-SSRC-23 dataset for the following: (a) Class 0 (Benign), (b) class 1 (brute force), (c) class 2 (DoS), (d) class 3 (information gathering), and (e) Class 4 (Mirai).

The local explainability analysis is expanded to all five TII-SSRC-23 traffic types in Figure 8. Dominant contributions of classes are present, which helps the interpretation of different combinations of traffic attributes in the network used to recognize benign, brute force, DoS, information gathering, and Mirai behavior. This is better than a single over-dominating variable-based model as it will be stronger under varying conditions of network.

There is also a practical benefit to the local explanations for the analyst. They prove to be more useful than the traditional explanations. Security operators can review the flow data characteristics that were most significant to an alert and do a side-by-side analysis to see if they resemble a characteristic of an attack. These explanations can be used to aid triage and auditing, but should also include the predicted probability, provide a true label for evaluation examples, and clearly define the SHAP background distribution, as these selections will influence the numerical attribution.

To determine the most influential features on a global scale, a SHAP analysis was performed on each class and dataset, and the results are summarized in the five figures above: Figures 9–13. The packet size and timing features, flow characteristics, and port behavior features were the most regular among the most important predictors. The results are consistent with the domain knowledge in the sense that the model learned meaningful traffic patterns, not arbitrary correlations. The SHAP analysis thus contributes to transparency and increasing the confidence in the developed framework for security-critical applications. Moreover, the agreement of the important features identified by SHAP with the ones selected by the hybrid feature selection procedure further confirms the meaningful and domain-related network traffic patterns captured by the proposed framework.

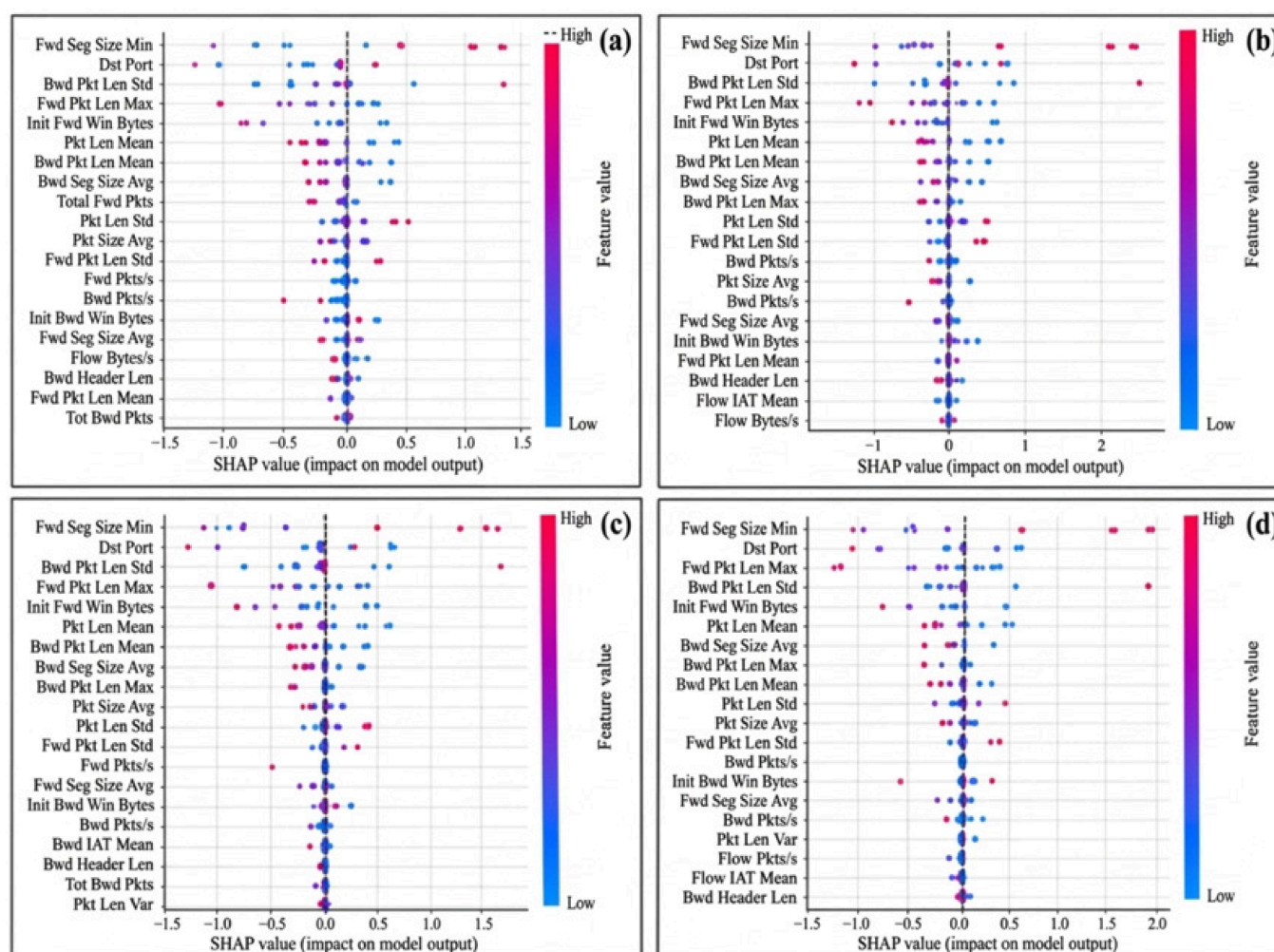


Figure 9. Global SHAP summary of Benign and three other classes in the CSE-CIC-IDS2018 dataset. Figure 9 presents the global SHAP explanations for the CSE-CIC-IDS2018 dataset. The subfigures provide class-specific global explanations for the following traffic categories: (a) class 0 (Benign), (b) class 1 (Bot), (c) class 2 (brute force-web), and (d) class 3 (brute force-XSS).

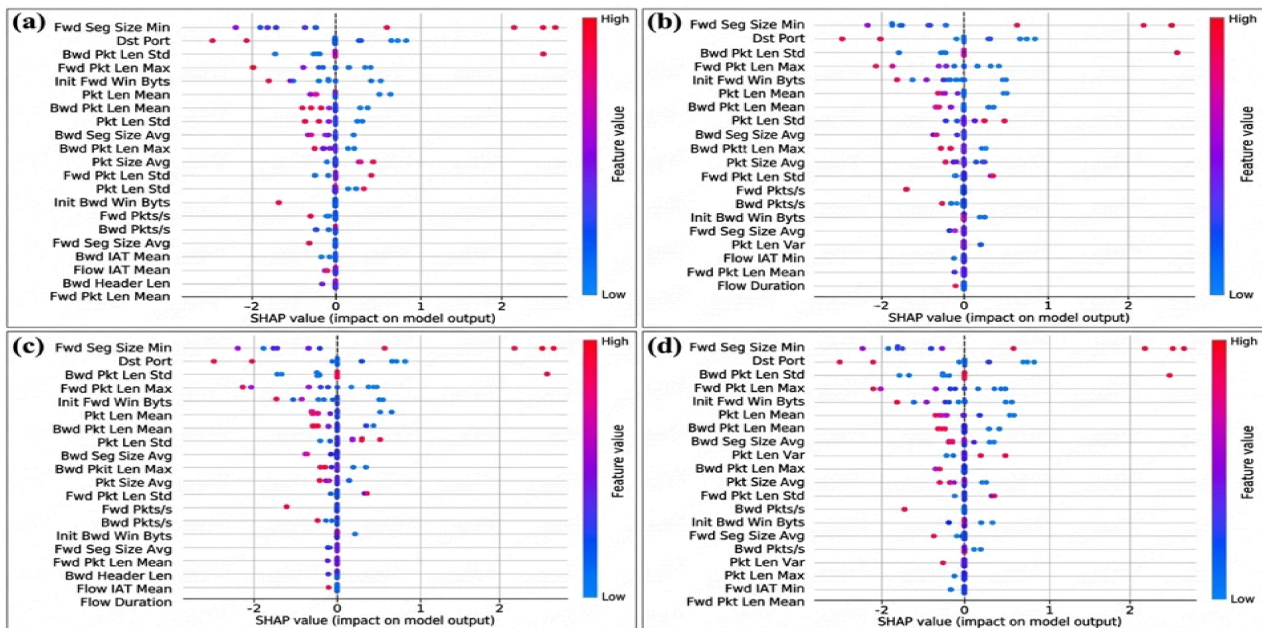


Figure 10. Global SHAP summary of four classes in the CSE-CIC-IDS2018 dataset. The subfigures of Figure 10 provide class-specific global explanations for the following traffic categories: (a) class 4 (DDoS attack-HOIC), (b) class 5 (DdoS attack-LOIC-UDP), (c) class 6 (DdoS attack-LOIC-HTTP), and (d) class 7 (DoS attack-GoldenEye).

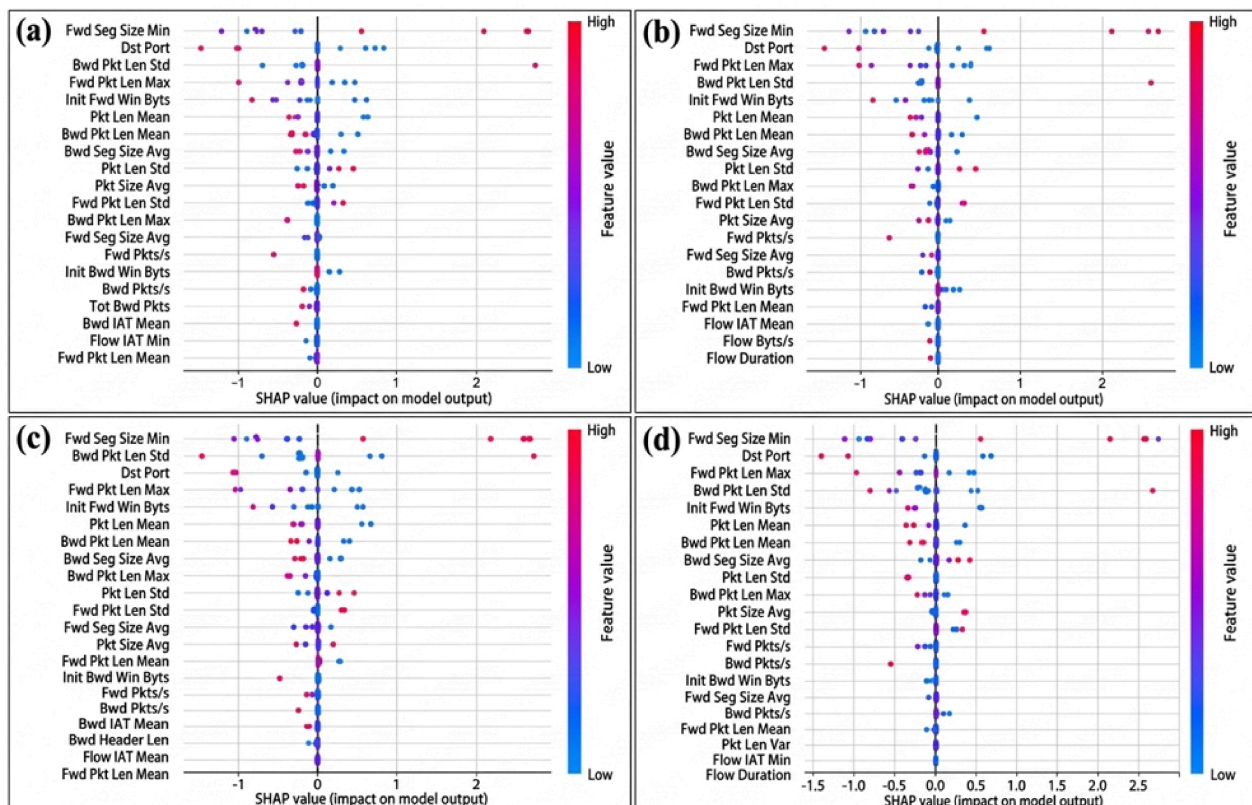


Figure 11. Global SHAP summary of DoS attack-Hulk and three other classes in the CSE-CIC-IDS2018 dataset. The subfigures of Figure 11 provide class-specific global explanations for the following traffic categories: (a) class 8 (DoS attack-Hulk), (b) class 9 (DoS attack-SlowHTTPTest), (c) class 10 (DoS attack-Slowloris), and (d) class 11 (FTP-brute force).

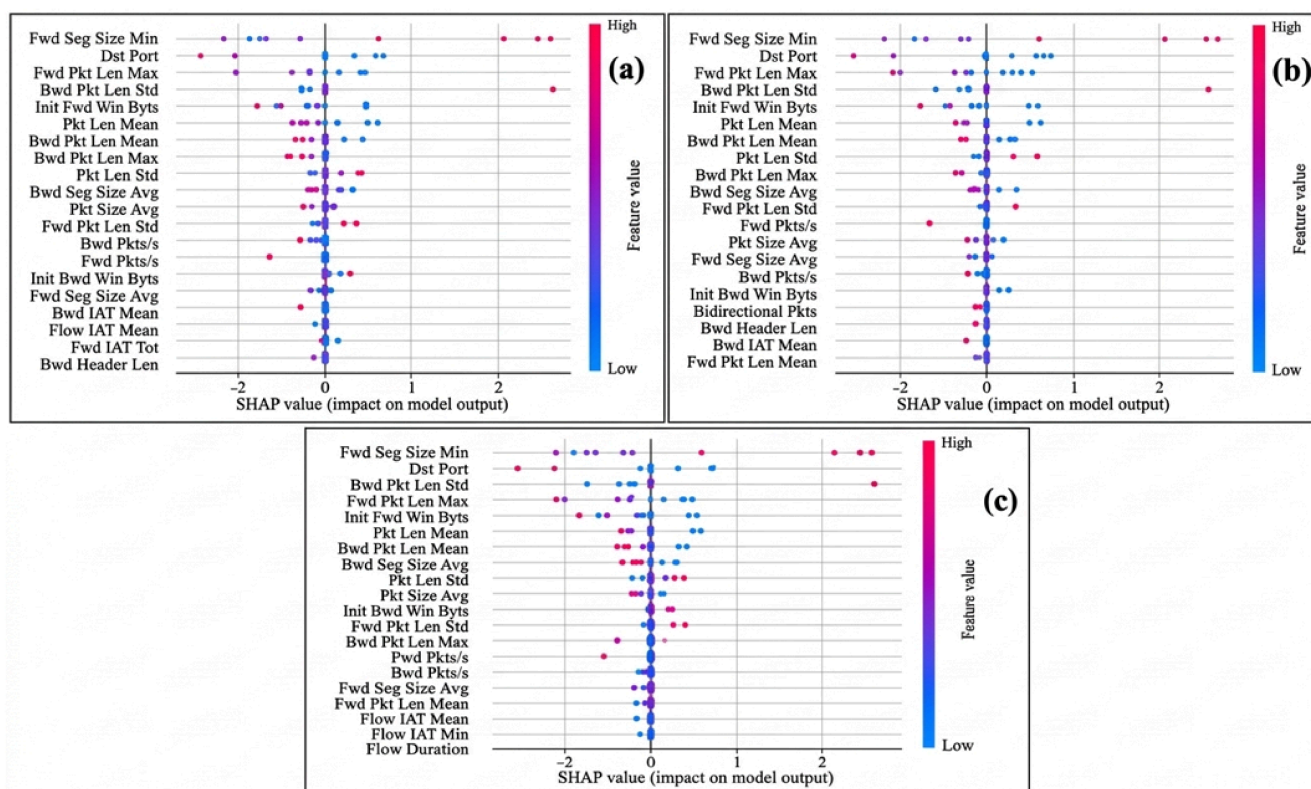


Figure 12. Global SHAP summary of the last three classes in the CSE-CIC-IDS2018 dataset. The subfigures of Figure 12 provide class-specific global explanations for the following traffic categories: (a) class 12 (infiltration), (b) class 13 (SQL injection), and (c) class 14 (SSH-brute force).

In a SHAP beeswarm summary, features are arranged by their mean absolute contribution. The direction in which the feature is affecting the model output is shown by the level of the graph, and the color of each point corresponds to the actual value of each feature in the dataset. The horizontal spread, then, reflects a range of importance and directionality of the observations. Broadly distributed features can affect different traffic instances in different ways and can interact with other variables. Packet-size statistics, packet-length measures, inter-arrival or flow-timing quantities, protocol/port indicators, and flow-level characteristics are consistently listed in the top 10 statistics consistently across the panels. The existence of these feature families in multiple classes of attacks is consistent with the physical structure of network attacks: flooding attacks change the rate and timing of attacks, brute-force behavior changes the pattern of connections and packets, and scanning/information-gathering behavior can change the pattern of ports and flows. This domain coherence argument reinforces the argument of interpretability, but does not prove causal validity alone.

Figure 10 illustrates that the feature can have different directional effects for different classes. This is to be expected in the multiclass classification case as a packet length or timing value that helps one attack family can also detract from another. The class-specific SHAP analysis therefore gives more information than a global ranking of the most important variables: it tells you which variables are useful to distinguish specific attack mechanisms.

Although there are a few features that have large SHAP values for a few observations, if the class support is not high, the model is unlikely to learn a stable decision region. Therefore, the meaning of explainability and predictive reliability is to be read together. A plausible SHAP pattern is not enough to offset the lack of confident recall based on a few test samples.

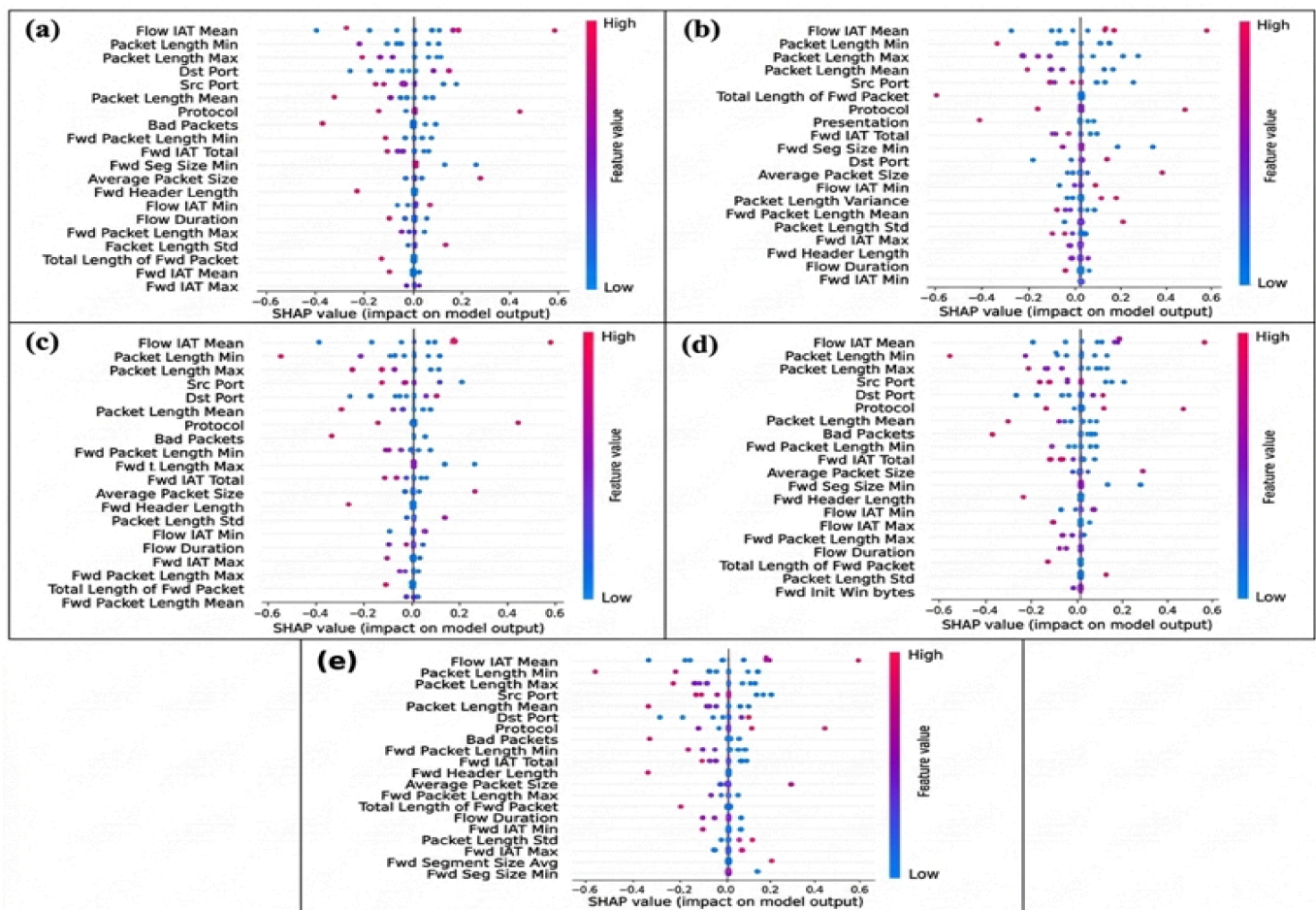


Figure 13. Global SHAP summary of all the traffic types in the TII-SSRC-23 dataset. The subfigures of Figure 13 provide class-specific global explanations for the following traffic categories: (a) class 0 (benign), (b) class 1 (brute force), (c) class 2 (DoS), (d) class 3 (information gathering), and (e) class 4 (Mirai). The selected features are in agreement with Table 4.

Infiltration, SQL injection and SSH-brute force are covered. In addition, the fact that the performance in SSH-brute force is good, while the others are unstable, is an example that rarity is not the only factor for difficulty. Separability can depend upon feature overlap, intra-class heterogeneity, attack conditions, and the extent to which an attack alters the flow statistics that are visible. This observation makes the class-aware feature selection and data enrichment more logical than using only feature importance across the entire dataset.

5. Conclusions and Future Research Directions

The findings of this study demonstrate that combining information gain (IG) with a genetic algorithm (GA)-based wrapper feature selection technique can substantially reduce the input feature space while maintaining strong intrusion detection performance. On CSE-CIC-IDS2018, the proposed IG²AFS-DNN framework retained 23 of 80 features and achieved a macro F1 of 0.833 and an MCC of 0.972. On TII-SSRC-23, it retained 15 of 86 features and achieved a macro F1 of 0.992 and an MCC of 0.999. The ablation results further demonstrate the complementary roles of the two feature selection stages: IG provides an efficient preliminary relevance filter, while the GA further refines the feature subset by considering multivariate predictive performance.

Comparative experiments showed that the proposed IG²AFS-DNN framework achieved competitive performance relative to the evaluated linear and probabilistic baselines and

remained competitive with strong tree-based ensemble models. The paired significance analysis provided statistical support for several observed differences, while non-significant comparisons indicated that the proposed framework did not consistently outperform all baseline models. Therefore, its principal contribution lies in achieving a favorable balance among dimensionality reduction, predictive performance, and SHAP-based explainability rather than demonstrating uniform superiority across all evaluated models.

The explainability analysis provides additional insight into the model's predictions. Local SHAP explanations illustrate feature contributions to individual predictions, while global class-wise SHAP summaries identify recurring influential features associated with packet, timing, flow, and port characteristics. The correspondence between features retained by the hybrid feature selection procedure and those identified as important by SHAP provides additional evidence that the model relies on relevant network traffic characteristics. However, SHAP explanations represent model associations and should not be interpreted as evidence of causal relationships.

An important limitation remains the detection of minority classes in CSE-CIC-IDS2018. The very small support for some attack classes results in unstable class-wise performance estimates and may bias the model toward majority classes. Future studies should therefore investigate class-aware or multi-objective feature selection, focal or cost-sensitive loss functions, carefully validated oversampling and data-augmentation techniques, threshold optimization, and few-shot learning approaches for rare attacks. Further evaluation should also incorporate repeated stratified or temporally separated splits, class-specific confidence intervals, calibration measures, and explicit data leakage and duplicate overlap audits.

Cross-dataset generalization and concept drift also warrant further investigation. Training on one dataset and evaluating on another, as well as testing on later temporal windows without retraining, would provide stronger evidence of generalization than evaluation based solely on random within-dataset partitions. In addition, reporting model size, training time, feature extraction overhead, inference latency, throughput, and energy consumption would enable a more comprehensive assessment of whether the reduced feature representation provides measurable deployment advantages in edge environments and high-throughput network security applications.

Overall, the results provide evidence that IG²AFS-DNN is a promising explainable intrusion detection framework that offers a useful trade-off among feature compactness, predictive performance, and interpretability across the two evaluated datasets. Further work should prioritize minority class robustness, reproducibility, explicit leakage assessment, cross-dataset validation, and evaluation under realistic temporal and operational conditions to establish the framework's broader applicability to real-world intrusion detection.

Author Contributions: Conceptualization, M.O.O. and Y.S.; methodology, M.O.O.; software, M.O.O.; validation, Y.S., M.O.O. and D.M.; formal analysis, M.O.O.; investigation, M.O.O., Y.S. and D.M.; resources, M.O.O.; data curation, M.O.O.; writing—original draft preparation, M.O.O.; writing—review and editing, Y.S. and D.M.; visualization, M.O.O.; supervision, Y.S. and DM.; project administration, Y.S.; funding acquisition, Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Research Foundation (NRF) of South Africa under Grant Nos. CPRR240327211161 and CHN231229201835.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this study are publicly available, and the original data source has been appropriately cited in the manuscript.

Acknowledgments: The authors acknowledge the use of ChatGPT (GPT-6, OpenAI) for assistance with text generation, language refinement, grammatical corrections, and enhancement of the visual clarity. All AI-assisted content was critically reviewed, edited, and verified by the authors, who take full responsibility for the accuracy, originality, and integrity of the final manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Singh, N.; Buyya, R.; Kim, H. Securing Cloud-Based Internet of Things: Challenges and Mitigations. *Sensors* **2025**, *25*, 79. [\[CrossRef\]](#) [\[PubMed\]](#)
2. Zhang, C.; Jia, D.; Wang, L.; Wang, W.; Liu, F.; Yang, A. Comparative research on network intrusion detection methods based on machine learning. *Comput. Secur.* **2022**, *121*, 102861. [\[CrossRef\]](#)
3. Admass, W.S.; Munaye, Y.Y.; Diro, A.A. Cyber security: State of the art, challenges and future directions. *Cyber Secur. Appl.* **2024**, *2*, 100031. [\[CrossRef\]](#)
4. Armoogum, S.; Armoogum, V.; Chandra, A.; Dewi, D.A.; Kurniawan, T.B.; Bappoo, S.; Salikon, M.Z.M.; Alanda, A. A Comprehensive Review of Cyber Hygiene Practices in the Workplace for Enhanced Digital Security. *Int. J. Inform. Vis.* **2025**, *9*, 137–145. [\[CrossRef\]](#)
5. Rathod, T.; Jadav, N.K.; Tanwar, S.; Alabdulatif, A.; Garg, D.; Singh, A. A comprehensive survey on social engineering attacks, countermeasures, case study, and research challenges. *Inf. Process. Manag.* **2025**, *62*, 103928. [\[CrossRef\]](#)
6. Roesch, M. Snort-Lightweight Intrusion Detection for Networks. In Proceedings of the LISA '99: 13th Systems Administration Conference, Seattle, WA, USA, 7–12 November 1999; pp. 7–12.
7. Biondi, P.S.A. Suricata: Next-Generation Intrusion Detection and Prevention System. In Proceedings of the 2012 Annual Computer Security Applications Conference (ACSAC '12), Orlando, FL, USA, 3–7 December 2012; pp. 95–104.
8. Yin, Y.; Jang-Jaccard, J.; Xu, W.; Singh, A.; Zhu, J.; Sabrina, F.; Kwak, J. IGRF-RFE: A hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset. *J. Big Data* **2023**, *10*, 15. [\[CrossRef\]](#)
9. Sonal; Deswal, S. Enhancing IoT gateway management security: A deep learning based framework for intrusion detection and threat mitigation. *Int. J. Inf. Technol.* **2025**, *17*, 3695–3705. [\[CrossRef\]](#)
10. Vinayakumar, R.; Soman, K.P.; Poornachandran, P. Applying convolutional neural network for network intrusion detection. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI, Udipi, India, 13–16 September 2017; pp. 1222–1228. [\[CrossRef\]](#)
11. Onuorah, M.O.; Yanxia, S.; Mashao, D. Deep Learning-Based Intrusion Detection System: Embracing Long Short-Term Memory (LSTM) and Roughly Balanced Bagging Synergies. *Intel. Artif.* **2025**, *28*, 40–65. [\[CrossRef\]](#)
12. Kasongo, S.M.; Sun, Y. A deep learning method with wrapper based feature extraction for wireless intrusion detection system. *Comput. Secur.* **2020**, *92*, 101752. [\[CrossRef\]](#)
13. Bakır, H.; Ceviz, Ö. Empirical Enhancement of Intrusion Detection Systems: A Comprehensive Approach with Genetic Algorithm-based Hyperparameter Tuning and Hybrid Feature Selection. *Arab. J. Sci. Eng.* **2024**, *49*, 13025–13043. [\[CrossRef\]](#)
14. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy*; SciTePress: Setubal, Portugal, 2018; pp. 108–116. [\[CrossRef\]](#)
15. Herzalla, D.; Lunardi, W.T.; Andreoni, M. TII-SSRC-23 Dataset: Typological Exploration of Diverse Traffic Patterns for Intrusion Detection. *IEEE Access* **2023**, *11*, 118577–118594. [\[CrossRef\]](#)
16. Rani R, H.J.; Barve, A.; Malviya, A.; Ranjan, V.; Jeet, R.; Bhosle, N. Enhancing detection rates in intrusion detection systems using fuzzy integration and computational intelligence. *Comput. Secur.* **2025**, *157*, 104577. [\[CrossRef\]](#)
17. Vinayakumar, R.; Alazab, M.; Soman, K.P.; Poornachandran, P.; Al-Nemrat, A.; Venkatraman, S. Deep Learning Approach for Intelligent Intrusion Detection System. *IEEE Access* **2019**, *7*, 41525–41550. [\[CrossRef\]](#)
18. Zhao, R.; Mu, Y.; Zou, L.; Wen, X. A Hybrid Intrusion Detection System Based on Feature Selection and Weighted Stacking Classifier. *IEEE Access* **2022**, *10*, 71414–71426. [\[CrossRef\]](#)
19. Vibhute, A.D.; Nakum, V. Deep learning-based network anomaly detection and classification in an imbalanced cloud environment. In *Procedia Computer Science*; Elsevier B.V.: Amsterdam, The Netherlands, 2024; pp. 1636–1645. [\[CrossRef\]](#)
20. Ciaramella, G.; Martinelli, F.; Santone, A.; Mercaldo, F. A method for smart grid intrusion detection through explainable deep learning. *J. Comput. Virol. Hacking Tech.* **2025**, *21*, 9. [\[CrossRef\]](#)
21. Tirumanadham, N.S.M.K.; Thaiyalnayaki, S.; Sriram, M. Improving predictive performance in e-learning through hybrid 2-tier feature selection and hyper parameter-optimized 3-tier ensemble modeling. *Int. J. Inf. Technol.* **2024**, *16*, 5429–5456. [\[CrossRef\]](#)
22. Ayad, A.G.; Sakr, N.A.; Hikal, N.A. A hybrid approach for efficient feature selection in anomaly intrusion detection for IoT networks. *J. Supercomput.* **2024**, *80*, 26942–26984. [\[CrossRef\]](#)

23. Wali, S.; Farrukh, Y.A.; Khan, I. Explainable AI and Random Forest based reliable intrusion detection system. *Comput. Secur.* **2025**, *157*, 104542. [[CrossRef](#)]
24. Fatema, K.; Dey, S.K.; Anannya, M.; Khan, R.T.; Rashid, M.M.; Su, C.; Mazumder, R. Federated XAI IDS: An Explainable and Safeguarding Privacy Approach to Detect Intrusion Combining Federated Learning and SHAP †. *Future Internet* **2025**, *17*, 234. [[CrossRef](#)]
25. Naif Alatawi, M. Enhancing Intrusion Detection Systems With Advanced Machine Learning Techniques: An Ensemble and Explainable Artificial Intelligence (AI) Approach. *Secur. Priv.* **2025**, *8*, e496. [[CrossRef](#)]
26. Hariharan, S.; Rejimol Robinson, R.R.; Prasad, R.R.; Thomas, C.; Balakrishnan, N. XAI for Intrusion Detection System: Comparing Explanations Based on Global and Local Scope. *J. Comput. Virol. Hack. Tech.* **2023**, *19*, 217–239. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.