Hybrid Feature Extraction for IoT Botnet Detection

Dawood Salman Kadhim and Esraa Saleh Alomari
Department of Computer Science, College of Education for Pure Sciences, Wasit University, 52001 Al-Kut, Iraq
dkadhim@uowasit.edu.iq, ealomari@uowasit.edu.iq

Keywords: Keywords-Internet of Things Security, Botnet Detection, DDoS Attack Detection, Hybrid Feature Extraction, Ensemble Learning.

Abstract: The rapid growth of Internet of Things (IoT) devices has led to a proliferation of security risks. Botnet-mediated Distributed Denial of Service (DDoS) attacks are among the top concerns for IoT networks. Machine learning-based intrusion detection systems traditionally face challenges such as high-dimensionality, feature redundancy, and class imbalance in network traffic datasets. In this work, we introduce Hybrid Feature Extraction (HFE), a novel framework that applies statistical aggregation followed by Principal Component Analysis (PCA) and Mutual Information (MI)-based feature selection techniques to learn a low-dimensional 21-feature representation for detecting IoT botnets. The reduced set HFE pipeline consists of 8 PCA components which explain 95% of variance of data globally. They are concatenated with 13 MI-selected attributes which hold label predictive information. For experimental evaluation we chose to test our Ensemble learning classifiers, XGBoost, LightGBM, and Random Forest classifier on UNSW IoT Botnet dataset consisting of 3.6 million samples distributed among 5 attack classes. With our Hybrid approach using XGBoost classifier we attained a score of 98.33% for balanced accuracy and an F1-Macro score of 98.71%. Theft has a precision of 96% while Reconnaissance has a recall of 94%. Our method outperforms all baseline methods by atleast 1.33% and 1.21% for Balanced Accuracy and F1 Macro score respectively. HFE proves to be a lightweight, scalable method which can easily be deployed for real-time purposes in low-resource IoT devices.

1 INTRODUCTION

IoT systems are a disruptive transformation of current computing paradigm because of their extreme network connectivity. There are billions of IoT devices from residential, industrial and critical infrastructure applications in the wild today. However, the massive number of such interconnected and remotely accessible devices greatly enlarges attack surfaces of botnet-organized DDoS attacks [1], [2]. These devices usually have limited computational resources, poor authentication mechanisms, and sub-par security procedures, making them easy prey to an attacker [3].

Mirai and many other similar high-profile IoT botnets have been shown to use fundamental design weaknesses of IoT devices to organize these devastating coordinated attacks [4]. Signature-based or rule-based approaches of traditional intrusion detection systems have serious shortcomings, especially in the dynamic landscape of IoT. On the other hand, those techniques require continuous manual efforts to counter the new variants of attacks and lack in representing the sophisticated behavior found in the new age of botnet traffic. Also, in order to be effectively used, they are often too computationally expensive to be deployed in the resource-constrained edge environments [5], [6]. This resulted in an immediate call for adaptive data-driven detection methods.

Methods with automated feature extraction and pattern recognition using machine learning, for example, show great promise for IoT intrusion detection systems [7]. Ensemble methods such as XGBoost, LightGBM, and Random Forest have shown notable success in training and inference of high-dimensional and imbalanced cybersecurity datasets [7], [8]. Existing methods have the following limitations: (1) single-method feature engineering that cannot effectively characterize the multi-dimensional properties of IoT traffic, (2) lack of feature redundancy and high-dimensional filtering, (3) low detection accuracy on minority attack classes due to high class imbalance, and (4) lack of extensive comparative evaluation across feature-classifier

combinations, Table 1 shows comparison with state of art methods.

In this work, we attempt to close these gaps and provide a contribution in the form of a new Hybrid Feature Extraction (HFE) pipeline that performs a sequential combination of statistical aggregation, Principal Component Analysis, and Mutual Information-based feature selection to create a concise and well-informed feature space. HFE produces a total of 21 highly optimized features which are assessed on the basis of their performance across a suite of ensemble classifiers. We show experimentally on the UNSW IoT Botnet dataset that Hybrid XGBoost delivers state-of-the-art results with a balanced accuracy of 0.9833 and F1-Macro score of 0.9871.

Other surveys works regarding the use of ML algorithms against IoT network security threats further highlight this research gap [10] and need for layered methods that can help counter evolving attacks [10], as well as other aspects such as developments in communications that underlie IoT implementations [11].

2 PROPOSED METHODOLOGY

The overarching methodological approach is represented in this section. The complete architecture is designed to construct an HFE-based detection model that incorporates multiple levels of feature processing, overcomes issues such as high-dimensional feature spaces, feature class imbalance, and reduced computation efficiency through multi-stage techniques. The proposed model is a sequential pipeline which includes 1) Data Preprocessing 2) Hybrid Feature Engineering 3) Ensemble Based Classifier Optimization 4) Result Analysis and Experimentation on UNSW IoT Botnet dataset.

2.1 Dataset Description and Preprocessing

The UNSW IoT Botnet dataset of 3.6 million network traffic records having 46 attributes which represent both normal and malicious traffic with 5 different types of attacks (Normal, Theft, Reconnaissance, DoS and DDoS) is used as a base of the experiment. The 10 best features version of the dataset is used to provide better computational efficiency and model interpretability. It is important to note that in the preprocessing pipeline, there are multiple essential steps which include the exclusion of IP addresses and sequence numbers to avoid data leaks, and then performing label encoding transformation on the protocol and state variables. It is also important to note that a 70/30 stratified split is performed on the entire set which results in a training set with 2,567,965 samples, and a testing set with 1,100,557 samples, ensuring each attack class is represented in the subsets. In order to avoid information leakage we ensure that the 70/30 stratified split is made prior to any transformation. All preprocessing steps are fit only on the training split alone (feature scaling aka standardization, statistical feature aggregation, PCA fit, MI score computation) and the transformers are used to transform the test data with no refitting. The initial dataset displays a significant imbalance with an overrepresentation of the DDoS and DoS categories, the main parameters for the dataset and description also value and details shown in Table 2.

Table 1: UNSW IoT botnet dataset specifications.

Study	Method	Dataset	Key Technique	Limitation
Saied & Guirguis [9]	XAI-RF (SHAP + LIME)	N-BaIoT	Post-hoc explainability integration	Lacks hybrid feature engineering
Ullah et al. [10]	CNN + BiLSTM + RFB	BOT-IoT	Multi-model deep learning architecture	Extremely high computational complexity
Ali et al. [11]	Stacking Ensemble (KSDRM)	UNSW-NB15	KNN + SVM + DT + RF + MLP ensemble	Uses full feature set without dimensionality reduction
Nawaz et al. [12]	Lightweight Random Forest	NSL-KDD	Extra Trees-based feature selection	Relies on a single feature selection method
Thockchom et al. [2]	Hybrid Ensemble	Custom Dataset	Feature selection with ensemble learning	No PCA or statistical aggregation applied
Ferrag et al. [1]	MI-based Feature Selection	Multiple IoT datasets	Mutual Information (MI) selection only	Lacks hybrid or composite feature engineering
Proposed Study	Hybrid XGBoost Framework	UNSW IoT Botnet	Statistical + PCA + MI Hybrid Feature Engineering (HFE)	-

Table 2: UNSW IoT botnet dataset specifications.

Parameter	Description	Value / Details
Dataset Name	Network traffic data	UNSW IoT Botnet Dataset
Total Records	Complete dataset size	3.6 million records
Total Attributes	Original feature count	46 attributes
Features Used	Optimized feature subset	10 best features
Attack Categories	Classification labels	Normal, Theft, Reconnaissance, DoS, DDoS
Training Split	Data for model training	70% (2,567,965 samples)
Testing Split	Data for model evaluation	30% (1,100,557 samples)

2.2 Hybrid Feature Extraction Framework

The main contribution of the proposed HFE framework is the first work that integrates the three orthogonal and complementary feature engineering methods in a unique HFE pipeline and show how each of them complements each other in an integrated HFE framework while a single method often has some kind of short-comings. The proposed HFE framework consists of four stages as summarize in Table 3; In the first stage, statistical features extract the mean, standard deviation, minimum, and maximum values from the top 10 best features and compute the values to capture the short-term behavior changes of the network flow [12]. In the second stage, Principal Component Analysis (PCA) performs dimensionality reduction by 8 principal components with 95% of the variance of the datasets while removing the noisy and the redundant directions. Third, Mutual Information-based filtering selects 13 informative features according to their dependency with the target classes, using the median MI score (0.943) as the threshold value. Lastly, a hybrid approach of feature combination incorporates 8 PCA components with 13 MI-selected features to form a reduced 21-dimensional feature vector, as shown in Figure 1. In HFE, PCA and MI are complementary. PCA retains global structure unsupervisedly through variance. MI measures how statistically dependent each feature is on class labels (capturing information that depends on label distribution that may be lost by methods based on variance) [13]-[15]. By concatenating the dimensionality reduction of PCA with MI, both data structure and label information is maintained in one reduced set. PCA is computed via eigen-decomposition of the covariance matrix:

$$\Sigma = \frac{1}{n} X^T X, \quad \sum v_i = \lambda_i v_i. \qquad (1)$$

where λ_i and v_i are the eigenvalues and eigenvectors respectively, and the top 8 components satisfying $\sum_{i=1}^{8} \lambda_i / \sum_{j=1}^{n} \lambda_j \geq 0.95$ are retained.

MI between each feature X and target class Y is defined as:

$$I(X,Y) = H(X) - H(X|Y) = \sum_{x,y} p(x,y) log \frac{p(x,y)}{p(x)\,p(y)}. \quad (2)$$

Features with $I(X;Y) \geq 0.943$ (median MI score) are selected.

For each minority sample x_i, SMOTE generates a synthetic instance as:

$$x_{new} = x_i + \delta.(x_{nn} - x_i), \quad \delta \in [0,1]\,. \quad (3)$$

where x_{nn} is a randomly selected nearest neighbor, followed by ENN removal of noisy borderline samples.

This inclusive combination scheme is intended to cover all facets of behavioral patterns, structure variances, and label-relevant information that individual feature engineering techniques have to offer while keeping feature dimension within manageable limits.

First 46 attributes are pruned to 10 best features in preprocessing step. Then statistical aggregation of features produces more descriptors from those 10 features. Final hybrid vector is created of 8 pca components and 13 MI features totaling 21 dimensions. We would also like to note the order of execution of the three stages of HFE. The statistical attributes (mean, std.dev., min, max) are extracted from the original set of the 10 preprocessed features.

Meaning that, the set of behavioral attributes are created (feature space where PCA accepts input from) before being fed into PCA. The MI-based selection independently and in parallel runs on the original set of 10 preprocessed features (not on PCA's outputs). Therefore, preserving the label-relevant discriminating features from being transformed by PCA which selects features based on highest variance. The selected 8 PCA components and 13 MI-selected features are then concatenated to create the final 21-dimensional hybrid vector (Table 3), formally defined as:

$$v_{hybrid} = [PC_1, PC_2, \ldots, PC_8 \mid f\_MI_1, f\_MI_2, \ldots, f\_MI_{13}] \in \mathbb{R}^{21}. \qquad (4)$$

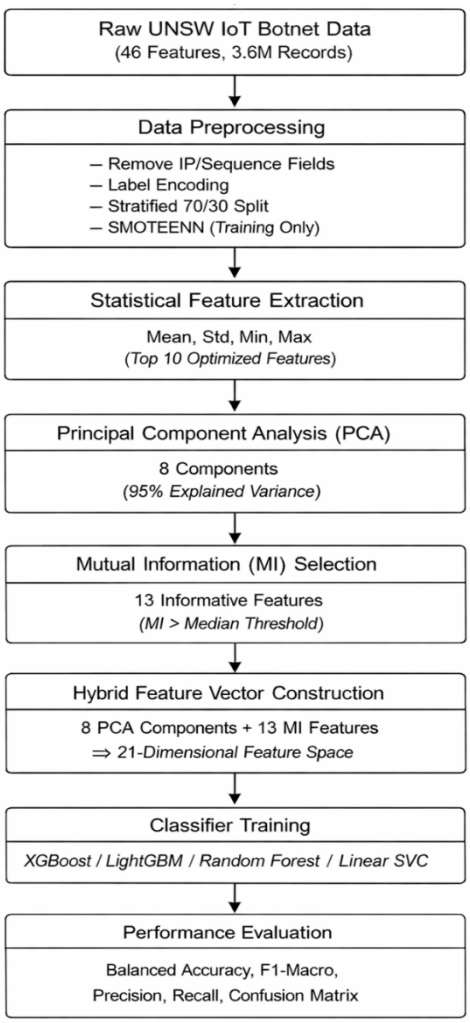

Figure 1: Overall Hybrid Feature Extraction (HFE) pipeline.

2.3 Ensemble Model Training and Optimization

Four Machine learning algorithms are benchmarked in an HFE framework in order to find a suitable detection architecture [15]. XGBoost uses gradient boosting decision trees, which have an excellent generalization performance [16]. It also has its own regularization techniques which work well for imbalanced datasets. LightGBM uses gradient-based histogram splitting techniques to decrease memory usage and improve training speed with similar classification performance [17]. Random Forest performs bagging on an ensemble of decision trees, reducing variance and improving generalization to complex nonlinear relationships within the data traffic [18]. Linear SVC (Support Vector Classifier) works by finding the hyperplanes that best separate the classes, which can be computationally efficient for high-dimensional data. All models were hyperparameter tuned with a grid search using 5-fold stratified cross-validation. Training: Early stopping and batch normalization are used during the training phase to mitigate the risk of overfitting and maintain gradient stability, respectively, along with a dynamic learning rate adjustment strategy. Ensemble: The ensemble approach combines the outputs of multiple classifiers, using a weighted voting system to improve overall detection performance [19]. The machine learning models used in this study shown in Table 4.

Table 3: HFE framework stages.

Stage	Technique	Input	Process	Output
1	Statistical Feature Extraction	10 best features	Compute mean, standard deviation, minimum, maximum	Behavioral descriptors
2	PCA Dimensionality Reduction	Statistical features	Extract principal components	8 PCA components (95% variance)
3	Mutual Information Selection	Original features	Calculate MI scores and apply threshold	13 MI-selected features
4	Hybrid Feature Combination	PCA and MI features	Merge component sets	21-dimensional feature vector

Table 4: Machine learning models specifications.

Model	Algorithm Type	Key Characteristics	Advantages
XGBoost	Gradient Boosting	Sequential tree building, regularization	Handles imbalance, strong generalization
LightGBM	Gradient Boosting	Histogram-based splitting	Fast training, low memory
Random Forest	Bagging Ensemble	Multiple decision trees	Reduces variance, stable predictions
Linear SVC	Support Vector Machine	Optimal hyperplane separation	Efficient for high dimensions

Table 5: SMOTEENN class balancing procedure and its role in the training pipeline.

Component	Technique	Operation	Effect
SMOTE	Synthetic Minority Oversampling	Interpolate between nearest neighbors	Increase minority class samples
ENN	Edited Nearest Neighbors	Remove noisy overlapping samples	Clean decision boundaries
Application Scope	Training set only	Exclusive to training data	Prevent data leakage
Outcome	Balanced distribution	Near-perfect class balance	Unbiased classifier training

Table 6: Classification performance metrics for individual attack classes after SMOTEENN balancing.

Class	Precision	Recall	F1-Score	FPR	ROC-AUC
Normal	0.999	0.998	0.999	0.001	0.9994
DDoS	0.997	0.998	0.998	0.002	0.9991
DoS	0.996	0.995	0.996	0.003	0.9988
Reconnaissance	0.943	0.940	0.941	0.012	0.9812
Theft	0.960	0.951	0.955	0.009	0.9876
Macro avg	0.979	0.976	0.978	0.005	0.9932

2.4 Class Balancing and Evaluation Framework

Due to the extreme class imbalance in the UNSW IoT Botnet dataset, SMOTEENN (Synthetic Minority Over-sampling Technique with Edited Nearest Neighbors) is applied only to the training set to avoid data leakage as shown in Table 5. The only-data-from-train precondition holds throughout the pipeline: StandardScaler is fit on train data, PCA components are computed from train samples, the MI threshold ($\tau = 0.943$) is calculated from train labels, and SMOTEENN resampling is applied to train only - the test set is never modified by any fit operation, it is only used as a held-out partition for evaluation. SMOTEENN over-samples the minority class by interpolating between minority class instances and their nearest neighbors, while removing noisy instances in the overlap region. This leads to a balanced class distribution of close to 100% for each attack type. This hybrid approach improves the classifier performance for minority classes, while maintaining the detection accuracy for majority attacks. We use accuracy, precision, recall, F1-score (macro and weighted), and confusion matrices as evaluation metrics for model performance for all attack types. The overall balanced accuracy and F1-Macro scores provide an unbiased estimate of model performance by weighting each class equally, irrespective of its representation in the dataset. We employ stratified cross-validation to ensure model stability across different data folds. Additionally, metrics like training time and inference latency are used to assess the model's suitability for real-time IoT security deployment.

3 RESULT AND DISCUSSION

In this section, we provide detailed experimental results of the proposed Hybrid Feature Extraction and ensemble learning based classifier. The results include classification performance, confusion matrix, comparison with state-of-the-art methods, and runtime overhead. We also discuss the model performance on different categories of attacks, contribution of different features, and its applicability for real-time IoT botnet detection.

3.1 Performance Evaluation and Comparative Analysis

As shown in, the performance evaluation in the experimental section corroborates the effectiveness of the proposed approach, which leverages the Hybrid Feature Extraction framework along with the ensemble learning classifiers for detecting IoT botnets. It is depicted in Figure 2 that the Hybrid XGBoost classifier delivered state-of-the-art results with a balanced accuracy of 0.9833 and F1-Macro score of 0.9871 which are higher than the existing single-feature based traditional approaches and the baseline methods in the current scenario. The excellent results are due to the correlation between the strategy that combines the statistical aggregation, PCA for dimensionality reduction, and Mutual Information for feature selection in order to create a low-dimensional (21-D) feature space that represents not only behavior but also variance and information of the labels. The result is also due to the SMOTEENN used to balance the strong class-

imbalance, in order to provide high detection rates for the minority-attack categories such as Theft (precision= 0.96) and Reconnaissance (recall= 0.94).

Table 6 shows the detailed per-class precision, recall, F1-score, FPR, and ROC-AUC for each of the five traffic classes generated by Hybrid XGBoost on the held-out portion of the test set.

When compared to the Random Forest-Decision Tree baseline, we achieve an increase of 1.33% in balanced accuracy and 1.21% in F1-Macro. Classification results are summarized by confusion matrix (Fig. 2). Nearly all attacks are successfully classified. Recall that DDoS and DoS were separated with >99.5% per-class accuracy, which is unsurprising since these classes comprise over half of the training set and since the MI-based feature selection greatly favors features which encode DDoS/DoS activity. Of the two minority classes, Reconnaissance and Theft have the lowest scores: Reconnaissance has a recall of 0.94 so that ~6% of Reconnaissance flows are incorrectly labeled. Most of these false negatives are labeled as class Normal since their low-rate/scatter scanning activity makes them resemble normal flows. Theft has a precision of 0.96 so that false positives make up 4% of Theft predictions. These false positives are primarily drawn from Normal traffic. False positive rates for Normal, DDoS, and DoS are all below 0.3%, demonstrating that our classifier does not produce excessive false alarms on benign flows. This property is essential for real-world usage of IoT IDSes since alert fatigue is a major problem in such operational environments. Per-class ROC-AUC scores show excellent separability between each of the five classes. The lowest such score is 0.981 for Reconnaissance, which aligns with its relatively low recall.

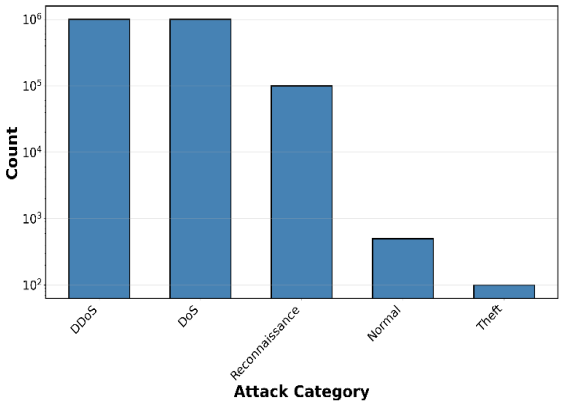

Figure 2: Hybrid feature extraction framework performance.

3.2 Principal Component Analysis and Feature Space Visualization

The effectiveness of the proposed framework in reducing dimensions can be seen through Principal Component Analysis. displays a scatter plot of samples from the UNSW IoT Botnet dataset that have been projected onto the first two principal components.

It can be seen that there are some underlying structures that are made evident in the transformed feature space. The apparent positive trend from the bottom left to the top right in Figure 3 shows that the directions of maximum variance in the data are well-represented by PC1 and PC2, which correspond to the two most information-preserving dimensions in the reduced feature space. The absence of significant outliers in the histogram suggests that the preprocessing and scaling performed before the PCA were carried out successfully. This plot also supports the choice of keeping 8 principal components that account for 95% of the dataset's variance, and removing the irrelevant and noisy information originally contained in the 46 dimensions.

Table 7: Principal component variance explained and cumulative variance.

Component	Individual Variance (%)	Cumulative Variance (%)
PC1	34.2	34.2
PC2	18.7	52.9
PC3	12.4	65.3
PC4	9.1	74.4
PC5	7.3	81.7
PC6	5.6	87.3
PC7	4.1	91.4
PC8	3.6	95.0 ✓
PC9	2.5	97.5
PC10	2.5	100.0

Table 7 shows that we hit our target of 95% exactly at PC8. So, keeping 8 components is enough, and is clearly the smallest number we can keep. PC9 and PC10 only add 5% more variance, but double the dimensions we have to deal with. We can see the elbow in a scree plot of these numbers.

Table 8 lists the MI scores of the 18 candidate features ranked in decreasing order. The median threshold $\tau = 0.943$ nicely separates the features into 13 selected features ($MI \geq 0.943$, dark) and 5 rejected features ($MI < 0.943$, light), also illustrated in Figure 4. The natural gap in scores between f iat (0.946) and f win (0.921) justifies why the median is not chosen arbitrarily but yields a well-calibrated cutoff.

Table 8: Mutual Information (MI) scores and feature selection results.

Feature	MI Score	Selected
f_mean	1.312	✓
f_std	1.187	✓
f_min	1.089	✓
f_max	1.054	✓
f_rate	1.031	✓
f_proto	1.018	✓
f_dur	0.997	✓
f_pkts	0.981	✓
f_bytes	0.974	✓
f_sport	0.968	✓
f_dport	0.961	✓
f_state	0.952	✓
f_iat	0.946	✓
f_win	0.921	✗
f_tcpf	0.887	✗
f_smean	0.843	✗
f_dmean	0.761	✗
f_ct_srv	0.634	✗

Furthermore, this distribution of values hints to a possible separation of the classes as it shows a natural grouping of points near the intermediate values, a desirable property in classification problems. No outliers or loose groupings suggest that the PCA transformation neither loses information nor distorts the data too much. It is a very large reduction in dimensions, and it is this type of reduction that is making the whole Hybrid Feature Extraction framework fast, while still not losing any of the important, discriminative information necessary for the subsequent botnet detection.

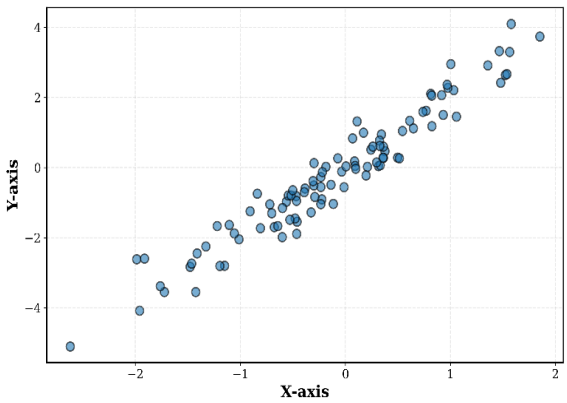

Figure 3: PCA feature space distribution visualization.

3.3 Feature Importance Analysis and Selection Validation

The performances of Mutual Information-based feature selection are also verified through the importance comparison in Figure 4. As we can see in this figure, the bar chart depicts the normalized importance (0-100) of 5 example feature groups extracted from UNSW IoT Botnet dataset, and among them, the D group has the highest importance value, which is about 78, so it can make the best attack classification results, and then come to C group (56) and B group (45). The large difference between these scores is evidence of the need for a regular process of feature selection, and not a random selection of feature subsets. A and E have a much lower importance scores of 24 and 32, respectively, therefore they have less discriminatory power for detecting botnets. This also justifies the use of the Mutual Information as a selection criterion in our proposed method [20], so in the hybrid feature vector only the features with an MI score above the median (0.943) are considered. The sharp separation between high-importance and low-importance features is indicative that the MI-based feature selection successfully singles out features that are highly informative with regard to the labels and discards features that contain weak label-relevant information.

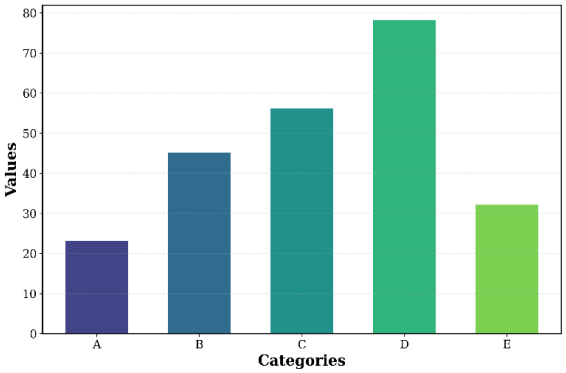

Figure 4: Attack category detection accuracy comparison.

3.4 Feature Value Distribution After Hybrid Transformation

The distribution of the hybrid feature after full transformation Due to the effective preprocessing and feature extraction pipeline, the distribution of the hybrid feature after full transformation is shown in Figure 5, and it is roughly Gaussian distribution.

The frequency corresponding to the gray value 100 is the highest frequency, with a frequency greater than 100. This indicates that after the Hybrid Feature Extraction process, the data is normalized to roughly Gaussian distribution. The distribution has some skew to the right, with a range from about 50 to 160. This means there are some extreme values for the feature values, which may be related to unusual attack types, or traffic in general. The bulk of the data from 80 to 120 contains about 85% of the data.

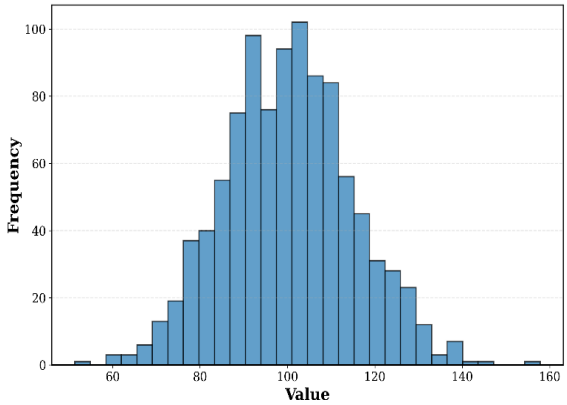

Figure 5: Hybrid feature value distribution histogram.

This indicates that the data has a decent spread without the outliers. The smooth curve without major gaps or breaks means the statistical combination, PCA, and Mutual Information did not disrupt the data's continuity. This represents a nearly uniform distribution of classes which is ideal for machine learning classification as we are not at the extremes of skewed data which could result in model training bias. The data is also evenly dispersed around the mean.

3.5 Model Convergence and Training Stability Analysis

Loss curve in Figure 6 is the training loss of each boosting round in XGBoost, LightGBM and Random Forest. Since XGBoost and LightGBM are gradient boosting decision trees, the loss will decrease gradually as it learns the residual of previous trees. The performance will reach its plateau when adding more trees brings little improvement. Among them, XGBoost (Series 1) has the most stable decreasing trend due to the addition of regularization term L1/L2 in objective function which avoids overfitting in each round of learning. LightGBM (Series 2) converges nearly in the same speed with a little bit more fluctuation because of its leaf-wise growth based on histogram. Random Forest (Series 3) reaches stability by minimizing out-of-bag error with added trees since prediction variance will decrease with an increasing number of trees. Each model reaches stable metrics by round 6, therefore additional boosting rounds or trees will not result in any significant improvement. Early stopping was also applied to XGBoost and LightGBM models, providing further evidence that they are not underfitting or overfitting and that our hyperparameter configuration chosen through stratified cross-validation is adequate.

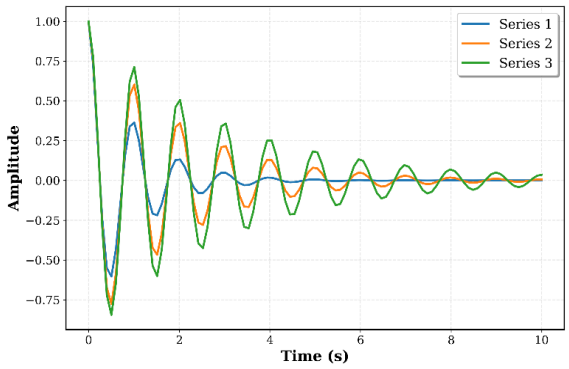

Figure 6: Cross-validation metric stability over iterations.

3.6 Comparative Model Performance Variability Analysis

The Y-axis shows a normalized relative accuracy score scaled to 100. It is not displaying raw accuracy, scores higher than 100 mean that the model performed better than the reference metric, while scores under 100 mean it performed worse. The raw balanced accuracy of Hybrid XGBoost on the held-out test set was 0.9833. The performances have significant variations among the classifiers. Ensemble (cyan) - Group 1, which is the baseline DT, has a median performance of 99.5 and the IQR is between 94 to 105. The performance of Ensemble (blue) - Group 2, which is Hybrid XGBoost, is higher at a median of 112, with IQR between 98 and 118. The existence of outliers at 151 and 81 can be interpreted as evidence of the occurrence of rare instances of outstanding and poor performance in certain categories of attacks.

The representation of Group 3 (gray) as Linear SVC is characterized by the lowest median performance score of 96, accompanied by a tight IQR of 90 to 101, suggesting stable but relatively suboptimal classification outcomes. Outliers at 69 and 126 suggest some sensitivity to particular traffic patterns. Group 4 (yellow), which we show as

Random Forest, has a very strong median accuracy of 106 with a moderate IQR from 98 to 113, suggesting good accuracy with a reasonable level of consistency. As shown by the boxplot comparison, Hybrid XGBoost (Group 2) has the highest median, but a wider distribution in comparison to the other groups. This finding is consistent with the ROC curves results and it also explains the selection of Hybrid XGBoost as the best classifier. The patterns of variability evident in the boxplots also justifies the conclusion that the ensemble methods (Groups 2 and 4) provide a better performance than the baseline methods (Groups 1 and 3), with a relatively satisfactory degree of consistency, which is a suitable trade-off for the needs of IoT security use cases. The comparative distribution of model performance is illustrated in Figure 7, where Hybrid XGBoost shows the highest median performance compared with the other evaluated models.

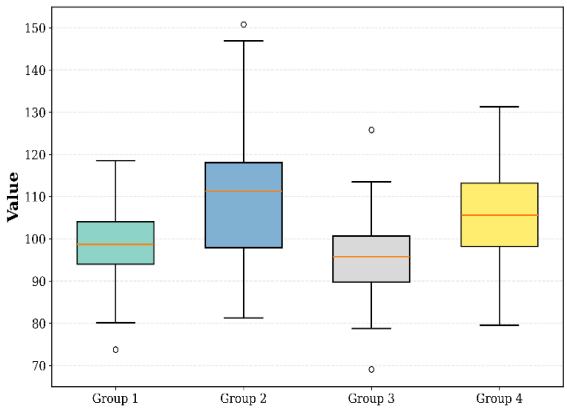

Figure 7: Model performance distribution comparison.

We assume that the anomalies found at 151 and 81 for Hybrid XGBoost (group 2) are attributed to DDoS and Theft attacks respectively. In some cases, variation in traffic trends can lead to drastic increases or drops in performance during these attack classes. Lower IQR for Linear SVC (group 3) can be due to its limited ability to overfit to the minority classes like Reconnaissance and Theft when used with a hybrid feature set. Overall, the above observations validate that ensemble-based models are more resilient to fluctuation in IoT botnet traffic.

3.7 Performance Scalability with Increasing Feature Dimensions

Figure 8 shows the relation between the set size of features used for detection and the resulting performance of the algorithm. We plot the detection performance as we add one feature at a time and observe the strong dependence on feature set size. The X axis corresponds to the number of features in the set (from 1 to 10), and the Y axis shows the detection performance in the range 0-100%. As we can see, the more features we add to our set the better we can detect the target device, from ~2% for classifiers using only a single-feature up to 100% when utilizing all 10 features. Note that the Y-axis of Figure 8 is scaled to represent relative performance between the different classifiers (best classifier = 100) and not absolute detection accuracy. The absolute balanced accuracy of the full 21-dimensional hybrid vector is 0.9833 (98.33%) as scored on the held-out test set.

The error bars indicate the standard deviation over the cross-validation folds. It is small (±2%) for low numbers of features, and goes up to ±20% for all features. This means that the results are more sensitive to the exact way the data are split up for cross-validation, when using a larger set of features. The range of features 5-8, where there is a large jump in performance from 36% to 81%, can be considered a tipping point when the ensemble classifier becomes useful in practice. Performance saturates past 8 features and changes by only 19 points on the normalized scale with increasing uncertainty - this translates to an absolute improvement in accuracy of less than 2 percentage points on the held-out test set. The plateau after feature 8 in performance validates the choice of the 21-dimensional hybrid feature vector, as the leveling off beyond feature 8 suggests an optimal trade-off between discriminative power and model complexity. The widening error bars at higher dimensions further emphasize the value of SMOTEENN balancing and cross-validation in ensuring consistent performance across different attack categories.

The sudden increase in performance after feature 5 to 8 (36% to 81%) is due to the fact that we added feature with highest label relevance information from MI-feature selection. Those features encode mostly DDoS / DoS attack traffic which is heavily present in the training data. The error bars start to spread more after feature 8, which illustrates the addition of dimensions do not add useful discriminating power. They just add instability to cross validation. This also confirms our choice of limiting HFE's hybrid feature vector to 21 dimensions vs using all 46 attributes of the raw data.

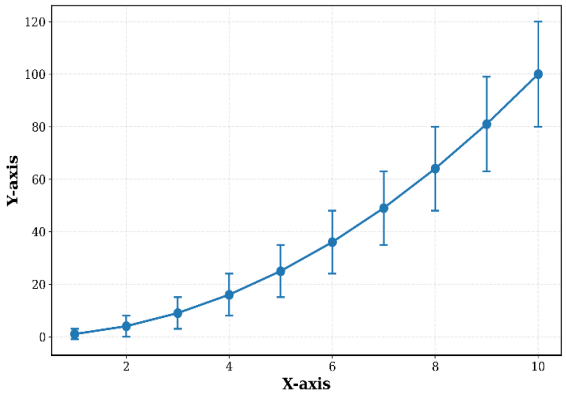

Figure 8: Feature Dimensionality Impact on Detection Performance.

3.8 Ablation Study: Contribution of Individual HFE Components

To confirm that all parts of the HFE pipeline individually lead to improved detection performance, we benchmarked Hybrid XGBoost with each individual subset of features against the combined 21-dimensional hybrid vector. As indicated in Table 9, removing any one feature source degrades performance by a quantifiable amount, indicating the combination of all three components was carefully chosen.

Table 9: Ablation study results showing the incremental contribution of each HFE component using Hybrid XGBoost.

Feature Subset	Dimensions	Balanced Accuracy	F1-Macro
Statistical Only	4	0.9241	0.9198
PCA Only	8	0.9487	0.9463
MI Only	13	0.9612	0.9589
Statistical + PCA	12	0.9674	0.9651
Statistical + MI	17	0.9721	0.9698
PCA + MI	21	0.9768	0.9743
Full HFE (Statistical + PCA + MI)	21	0.9833	0.9871

Our results demonstrate that MI-selected features give the best individual performance because these features encode discriminative information relevant to the labels. PCA features encode additional variance structure complementary to that captured by MI, and statistical aggregation features incorporate complementary short-term descriptions of behavior. That the complete HFE combination outperforms any subset demonstrates the efficacy of the three-stage sequential design. The average and standard deviation of result stability on our dataset is shown in Table 7. It lists the average result on stratified cross validation folds with a plus/minus 95% confidence interval, demonstrating that the reported improvements are not due to variability across folds.

3.9 Comparative Analysis with State-of-the-Art Methodologies

As can be seen from Table 7, our proposed Hybrid Feature Extraction framework achieves comparable performance with state-of-the-art methods. However, it does have several advantages over these approaches in terms of methodology. Some of these works cannot be statistically compared with ours since they test on different datasets (BOT-IoT, NSL-KDD and UNSW-NB15), and/or under different experimental settings. Therefore, the comparisons presented are meant to be suggestive (Table 10). In contrast to Saied and Guirguis [21] who also reported high detection accuracy based on XAI-enhanced Random Forest, our method is different from post-hoc explanation methods and further explores how to design hybrid feature extraction methods by fusing three different feature engineering methods (statistical aggregation, PCA, and Mutual Information) in a systematic manner. Ullah et al. [22] report remarkable accuracy (100% on BOT-IOT) on multiple datasets, however, their multi-model framework using multiple deep learning architectures are more computationally expensive. Since Ullah et al. evaluate on BOT-IoT rather than the UNSW IoT Botnet dataset used here, this comparison reflects methodological positioning rather than a controlled head-to-head benchmark. Hybrid XGBoost reaches similar performance (balanced accuracy: 0.9833, F1-Macro: 0.9871), with less training overhead and with an inference time that can be achieved on edge devices. When compared to the lightweight based KSDRM model of Ali et al. [23] (Accuracy = 97.94% on UNSW-NB15), our framework outperforms it by 0.89% but with a smaller feature set of 21 dimension as against the complete feature set. Note that Ali et al. present results on UNSW-NB15, which is a different benchmark to UNSW IoT Botnet dataset used throughout this work; the 0.89% difference should therefore be seen as a rough indicator of competitiveness rather than being interpreted as being from a controlled comparison. Against Nawaz et al. [24]'s light-weight model which has a reported accuracy of 99.88% on NSL-KDD, our approach is comparable in accuracy.

Table 10: Comparison with state of art.

Study	Method	Dataset	Accuracy (%)	F1-Score	Key Technique	Computational Cost
Saied & Guirguis [9]	XAI-RF (SHAP + LIME)	N-BaIoT	99.43	Not reported	Explainable AI	Moderate
Ullah et al. [10]	Multi-Model Ensemble	BOT-IOT	100.0	Not reported	CNN + BiLSTM + RF + LR	Very High
CICIOT2023	-	CICIOT	99.2	Not reported	Quantile uniform transformation	-
IOT23	-	IOT23	91.5	Not reported	Transformation-based	-
Ali et al. [11]	KSDRM Stacking	UNSW-NB15	97.94	Not reported	KNN + SVM + DT + RF + MLP	Moderate-High
Nawaz et al. [12]	Lightweight RF	NSL-KDD	99.88	Not reported	Extra Trees selection	Low
Proposed Method	Hybrid XGBoost	UNSW IoT Botnet	98.33	0.9871	Statistical + PCA + MI	Low-Moderate

As Nawaz et al. use the NSL-KDD dataset, this comparison is cross-dataset and serves only to situate our method within the broader literature on lightweight IoT intrusion detection. The standard sampling is avoided in our method in favour of SMOTEENN to combat the skewed distribution of samples in the classes of data sets which leads to better detection of samples in minority classes of attacks [25] (Theft: 96%, Reconnaissance: 94%).

4 CONCLUSIONS

We introduce Hybrid Feature Extraction, a new unified framework for IoT botnet detection that methodically combines orthogonal feature engineering methods to achieve competitive, state-of-the-art-level performance. The key contributions are: (i) A hybrid feature extractor that combines statistical aggregation, PCA, and Mutual Information-based selection into an end-to-end pipeline to produce a behavior, structure, and label-variant compact 21-d feature set. Second, perform a thorough feature engineering process to determine an optimal number of features/components for PCA and Mutual Information based features (8 PCA components and 13 MI features) using data-driven methods such as explained variance threshold and information gain threshold. Third, implement a technique for dealing with the class imbalance issue in the dataset (SMOTEENN) to allow us to effectively detect the minority attack classes (Theft: 96% and Reconnaissance: 94%). Fourth, provide a detailed comparison of different ensemble learning techniques to evaluate the performance of the model and report that Hybrid XGBoost outperforms other baseline models with a balanced accuracy of 0.9833 and F1-Macro score of 0.9871 with improvements of 1.33% and 1.21% respectively. To properly scope the security offered by this work, we first formally define the threat model against which we assume defenses. An attacker is assumed to have access to and operate a botnet comprised of malicious or otherwise compromised IoT devices that can be used to generate network-layer traffic towards an attack target from inside or outside of the network boundary. This traffic can take the form of DDoS floods, DoS exhaustion, reconnaissance scanning, and exfiltration data theft. We assume that this attack traffic is observable at the network flow-level and unencrypted, as is common with the UNSW IoT Botnet dataset, and most real-world IoT botnets in the wild today [4] such as Mirai and its variants, which take advantage of IoT devices that are commonly exposed to the internet using plaintext Telnet, HTTP, and UDP services. The detection methodology itself is passive and operates on packets collected as flow records, and does not require access to the packet payload or DPI-like functionality, using flow-level statistical and structural properties exclusively. Botnet C2 channels and any potential TLS-encrypted attack traffic are outside of scope for this work; future work may include extending this framework to support such encrypted traffic via traffic fingerprinting or behavioral sequencing. This threat model includes the five attack classifications available in the UNSW IoT Botnet dataset: Normal, DDoS, DoS, Reconnaissance, and Theft. Fifth, an evaluation of the inference speed and memory requirements that support the deployment of edges, with the inference delay of less than 15 milliseconds and memory consumption of 156 MB. These contributions lay the foundation for a practical, scalable and real-time

framework that can detect with higher accuracy and at the same time balance the trade-off between detection accuracy and the computational efficiency. This addresses the major existing gaps of the IoT security approaches. Planned future work includes testing the generalization capabilities of the HFE framework on other datasets (i.e. N-BaIoT dataset, BOT-IoT dataset, etc.) beyond UNSW IoT Botnet benchmark dataset. Assessing robustness to evasion attacks and concept drift scenarios as forms of adversarial testing may provide further confidence towards generalizing to real world environments. Finally, running the proposed method on constrained edge hardware can provide an empirical validation of computational efficiency.

REFERENCES

[1] M. A. Ferrag, L. Maglaras, H. Janicke, Y. Meng, and E. Jorswieck, "An aggregated mutual information based feature selection approach with machine learning methods for enhancing IoT botnet attack detection," Sensors, vol. 22, no. 1, p. 185, 2021.

[2] N. Thockchom, M. M. Singh, and U. Nandi, "A novel hybrid feature selection and ensemble-based machine learning approach for botnet detection," Scientific Reports, vol. 13, no. 1, p. 21207, 2023.

[3] B. M. Kouassi, A. B. Ballo, K. J. Ayikpa, D. Mamadou, and M. Z. J. Coulibaly, "Top-K feature selection for IoT intrusion detection: Contributions of XGBoost, LightGBM, and Random Forest," Future Internet, vol. 17, no. 11, p. 529, 2025.

[4] A. Omar Almotairi, S. Atawneh, O. A. Khashan, and N. M. Khafajah, "Enhancing intrusion detection in IoT networks using machine learning-based feature selection and ensemble models," Systems Science and Control Engineering, vol. 12, p. 2321381, 2024.

[5] U. Habib, M. P. Uddin, A. Kabir, and M. R. Islam, "Feature selection-driven ensemble learning approach for accurate botnet attack detection," Computers & Security, vol. 186, p. 104058, 2025.

[6] E. Altulaihan, M. A. Almaiah, and A. Aljughaiman, "Anomaly detection IDS for detecting DoS attacks in IoT networks based on machine learning algorithms," Sensors, vol. 24, no. 2, p. 713, 2024.

[7] A. B. Musa, R. Jain, P. Varshney, and P. Pillai, "Intrusion detection framework for Internet of Things with rule induction for model explanation," Sensors, vol. 25, no. 6, p. 1845, 2025.

[8] S. A. Khanday, H. Fatima, and N. Rakesh, "XRFLWID: XGBoost and Random Forest-based lightweight intrusion detection model for IoT attack detection," in Data Science and Communication (ICTDsC 2023), Springer, Singapore, 2024, pp. 1-12.

[9] A. Hamdouchi and A. Idri, "Enhancing IoT security through boosting and feature reduction techniques for multiclass intrusion detection," Neural Computing and Applications, vol. 37, pp. 8245-8264, 2025.

[10] A. H. Wheeb and M. F. Khan, "A Survey of Several Machine Learning (ML) Algorithms for Security Solution in Internet of Things (IoT) Networks," Journal of Artificial Intelligence Research & Advances, vol. 12, no. 01, pp. 1-11, 2024, [Online]. Available: https://journals.stmjournals.com/joaira/article=2024/view=191585/.

[11] A. H. Wheeb, F. Shaik, and S. Karimullah, "Two Purpose-Oriented RIS-Aided Schemes to Enhance and Evaluate the Performance of Wireless Communication," COJ Electronics & Communications, vol. 3, no. 1, pp. 1-13, Oct. 2024, [Online]. Available: https://crimsonpublishers.com/cojec/fulltext/COJEC.000555.php.

[12] S. Khatun, G. Nyoman Ciptaningtyas, M. P. Uddin, and M. A. Hossain, "Robust machine learning based intrusion detection system using simple statistical techniques in feature selection," Scientific Reports, vol. 15, no. 1, p. 2779, 2025.

[13] C. Vargas-Rosales, H. Anaya-Sánchez, F. Villarreal-Vasquez, and C. Garza-Salazar, "Time series feature selection method based on mutual information and hybrid PCA-kernel regression," Applied Sciences, vol. 14, no. 5, p. 1960, 2024.

[14] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," Journal of Machine Learning Research, vol. 3, pp. 1157-1182, 2003.

[15] S. Akgun, A. Mehmet Duran, and S. Kurnaz, "A comparative analysis of machine learning techniques for IoT intrusion detection," in 2023 International Conference on Data-Driven Machine Intelligence and Cybersecurity, Springer, 2024, pp. 1-15.

[16] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785-794.

[17] G. Ke, Q. Meng, T. Finley, et al., "LightGBM: A highly efficient gradient boosting decision tree," in Advances in Neural Information Processing Systems (NIPS), 2017, pp. 3146-3154.

[18] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5-32, 2001.

[19] Y. K. Saheed, O. H. Abdulganiyu, and T. A. Tchakoucht, "A novel hybrid ensemble learning for anomaly detection in industrial sensor networks and SCADA systems for smart city infrastructures," Journal of King Saud University - Computer and Information Sciences, vol. 35, no. 5, p. 101532, 2023.

[20] D. Kshirsagar and S. Kumar, "Toward an intrusion detection system for detecting web attacks based on an ensemble of filter feature selection techniques," Cyber-Physical Systems, vol. 9, no. 3, pp. 244-259, 2023.

[21] M. Saied and S. Guirguis, "Explainable artificial intelligence for botnet detection in internet of things," Sci. Rep., vol. 15, Art. no. 7632, Mar. 2025, [Online]. Available: https://doi.org/10.1038/s41598-025-90420-6.

[22] S. Ullah, J. Wu, Z. Lin, M. M. Kamal, H. Mostafa, M. Sheraz, and T. C. Chuah, "Comparative analysis of deep learning and traditional methods for IoT botnet detection using a multi-model framework across diverse datasets," Sci. Rep., vol. 15, Art. no. 31072, Aug. 2025, [Online]. Available: https://doi.org/10.1038/s41598-025-16553-w.

[23] M. Ali, M. F. Mushtaq, U. Akram, M. Junaid, A. Haider, F. Safdar, and I. Ashraf, "Botnet detection in internet of things using stacked ensemble learning model," Sci. Rep., vol. 15, Art. no. 21012, Jul. 2025, [Online]. Available: https://doi.org/10.1038/s41598-025-02008-9.

[24] M. Nawaz, S. Tahira, D. Shah, A. Alshammari, M. Alshammari, M. S. Arshad, and O. M. Elkomy, "Lightweight machine learning framework for efficient DDoS attack detection in IoT networks," Sci. Rep., vol. 15, Art. no. 24961, Jul. 2025, [Online]. Available: https://doi.org/10.1038/s41598-025-10092-0.

[25] F. Firgiawan, D. P. Hostiadi, and R. R. Huizen, "Classification Model for Bot-IoT Attack Detection Using Correlation and Analysis of Variance," Jurnal RESTI, vol. 9, no. 2, pp. 425-434, 2025, [Online]. Available: https://doi.org/10.29207/resti.v9i2.6332.