International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(19s) 2026 pp. 859-879
https://doi.org/10.70917/ijcisim-2026-5087
Article

A Context-Aware Genetic Algorithm Optimized Explainable Machine Learning Framework for Multi-Class IoT Botnet Detection Using the IoT-23 Dataset

Rintu Das[1], Vaskar Deka2*, GomTaye[3]

[1]Department of Information Technology, Gauhati University, Assam, India. rintu.das@gmail.com
2*Department of Information Technology, Gauhati University, Assam, India. vaskardeka@gauhati.ac.in
[3]Department of Computer Science and Engineering, Rajiv Gandhi University, Arunachal Pradesh-791 112 gomtaye@gmail.com

Abstract: The exponential growth of Internet of Things (IoT) devices has tremendously increased the attack surface of modern cyber-physical systems. Therefore, the detection of IoT botnets is an urgent challenge for cybersecurity. IoT environments are different from traditional enterprise networks. They are composed of heterogeneous and resource-constrained devices, and produce highly dynamic network traffic, which makes attack detection significantly more challenging. Current machine-learning-based intrusion detection systems suffer from class imbalance, redundant network features, limited context awareness, and poor interpretability. In this paper, we propose a context aware genetic algorithm optimized machine learning (CAGA-ML) framework for the explainable multi-class IoT botnet detection using the IoT-23 dataset. The proposed framework integrates a comprehensive pre-processing of data, adaptive hybrid class balancing, Mutual Information based feature filtering, DEAP based Genetic Algorithm based feature optimization, context aware K-means clustering and SHAP based explainability as one single intrusion detection pipeline. Network-flow features were extracted and optimized from the Zeek logs. Then, these features were used to train eight classical machine learning classifiers namely Decision Tree, Random Forest, Logistic Regression, Gaussian Naive Bayes, Linear support vector machine, K-Nearest Neighbors, CatBoost and XGBoost. To reduce the model selection bias and guarantee a reliable performance estimation nested cross-validation was used. The experimental results demonstrate that the proposed framework is able to effectively reduce the redundancy of features, improve the detection of minority class, and enhance the interpretability of model with computational efficiency.
Keywords: IoT-23, Internet of Things, Botnet Detection, Explainable AI, Genetic Algorithm, SHAP, Feature Selection, Machine Learning

1. Introduction

The Internet of Things (IoT) has changed the modern digital infrastructure, allowing billions of interconnected devices to communicate autonomously in healthcare, industrial automation, transportation, agriculture and smart cities [1]. IoT technologies provide new opportunities for intelligent automation, but they also expose networks to more sophisticated cyber-attacks [2]. IoT devices with limited processing power and heterogeneous communication protocols are attractive targets for malware, botnets, ransomware and distributed denial of service (DDoS) attacks [3]. One of the most significant security problems of these threats is the use of IoT botnets. Botnets such as Mirai, Gafgyt, Okiru, Torii and Muhstik leverage vulnerable IoT devices to launch coordinated attacks against internet infrastructure Hence, the early detection of a malicious behavior in a network is crucial to safeguard IoT ecosystems [4]. IoT-23 is one of the most widely adopted benchmark datasets in IoT intrusion detection research. Unlike synthetic datasets, IoT-23 is a collection of 23 real network capture scenarios (including 20 malicious botnet scenarios and 3 benign scenarios

from real IoT devices). The Zeek network monitoring framework processes network traffic to obtain flow-based features that can be used for machine-learning-based intrusion detection [5].

Machine learning has made great progress, but many challenges remain.

[1]. Attack types are heavily imbalanced by class

[2]. Highly correlated and redundant network flow features

[3]. No behavioral context information

[4]. Computational complexity of wrapper feature selection

[5]. Limited explainability of high-performing classifiers

To tackle these challenges, this research presents a context-aware genetic algorithm optimized machine learning (CAGA-ML) framework. The proposed method incorporates adaptive hybrid balancing, Mutual Information filtering, Genetic Algorithm feature optimization, context-aware clustering, nested cross-validation and SHAP explainability [6] in a unified intrusion detection pipeline tailored to the characteristics of the IoT-23 dataset.

The main contributions of this work are:

1. Context-aware explainable machine learning framework for IoT-23 botnet detection.
2. Adaptive Hybrid Balancing with Random Under-Sampling, Borderline-SMOTE and SMOTE.
3. Two-stage feature optimization based on Mutual Information and Genetic Algorithm searching.
4. Behavioral clustering with context-aware minibatch K-means.
5. General comparison of eight classical machine learning algorithms.
6. Explainability of transparent-attack classification via SHAP.

2. Related Work

The IoT-23 dataset is one of the most popular benchmark datasets for evaluating machine learning-based intrusion detection systems, because it comprises realistic botnet traffic captured from operational IoT devices. Recently, decision tree models, Random Forests, Support Vector Machines, XGBoost, LightGBM and deep learning architectures have been explored for botnet detection using IoT-23.[7], [8]

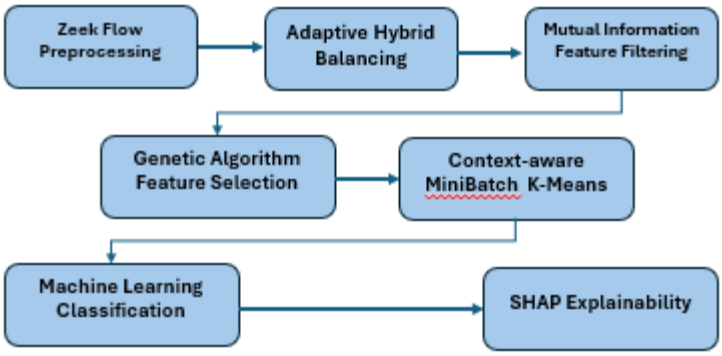
These models have been shown to successfully identify botnet activities in the dataset to a certain degree. Often, methods combining multiple models, such as Random Forests and gradient boosting algorithms like XGBoost and LightGBM, can achieve higher detection rates, as they are able to handle complex interactions between features. Moreover, deep learning techniques have been successfully used for automatically extracting features from raw network traffic data, to further enhance the detection performance[9]. The dimension of features is often reduced by using feature selection techniques such as Recursive Feature Elimination, Information Gain, Chi-Square, ReliefF, and Mutual Information[10]. Evolutionary optimization methods such as genetic algorithms, particle swarm optimization and gray wolf optimization have also been shown to provide promising results to identify compact feature subsets[11]. These techniques improve the efficiency of feature selection by effectively searching in large search space and avoiding local optima. To achieve better performance, hybrid algorithms of evolutionary algorithms with traditional feature selection methods have also been proposed. These integrative strategies exploit the strengths of both paradigms to identify highly relevant and non-redundant feature subsets[12]. Class imbalance is still a challenge in IoT-23 since benign traffic and dominant attack categories are substantially more frequent than the less frequent botnet behaviors. While SMOTE and ADASYN increase the number of minority class examples, over-sampling can lead to the generation of noisy synthetic samples. The hybrid balancing strategies that combine undersampling and oversampling have received increasing attention [13]. Hybrid methods combine the advantages of both techniques to provide a more balanced and representative dataset. These methods try to minimize the risk of overfitting and synthetic noise through

selective undersampling of the majority classes and careful oversampling of the minority classes. This balance improves the detection capability of machine learning model for rare botnet activities in IoT traffic [14]. Explainable Artificial Intelligence has recently become important in intrusion detection research. SHAP gives theoretically grounded local and global explanations of feature importance, which allows cybersecurity analysts to interpret model decisions and increase trust in automated detection systems [15]. However, existing IoT-23 studies typically optimize only isolated components of the intrusion-detection pipeline [5]. Only a few studies have incorporated adaptive balancing, evolutionary feature optimization, contextual feature engineering, rigorous nested cross-validation, and explainability [34, 35] into a single framework [16]. The proposed CAGA-ML architecture solves these problems.

CAGA-ML combines evolutionary algorithms with adaptive balancing techniques to dynamically optimize the feature subsets used in the training of the model. It uses contextual feature engineering to capture domain-specific patterns that are relevant to IoT botnet behaviors.[17] Furthermore, the framework applies nested cross-validation for a robust assessment of performance and incorporates explainability using SHAP for transparent insights into the model.

3. Proposed Framework

The proposed CAGA-ML framework consists of seven sequential stages.



1. Zeek Flow Preprocessing
2. Adaptive Hybrid Balancing
3. Mutual Information Feature Filtering
4. Genetic Algorithm Feature Selection
5. Context-aware MiniBatch K-Means
6. Machine Learning Classification
7. SHAP Explainability

4. Materials and Methods

The proposed framework is developed on the basis of the IoT-23 dataset which comprises 23 realistic network capture scenarios containing benign and malicious IoT traffic in Zeek flow records. The dataset includes different IoT botnet activities and can be evaluated in a realistic network environment. First, the raw data is preprocessed by normalizing labels, handling missing values, removing duplicate records, and encoding categorical Zeek attributes. The resultant data set is a clean and consistent set of features for analysis. To overcome the large class imbalance between benign traffic and multiple attack categories, while maintaining the underlying data distribution to ensure a balanced class representation, an Adaptive Hybrid Balancing (AHB) strategy is used. Then, we use Mutual

Information (MI) to remove irrelevant and redundant features, and then a Genetic Algorithm (GA) implemented with the DEAP framework to find the optimal subset of discriminative Zeek flow features.

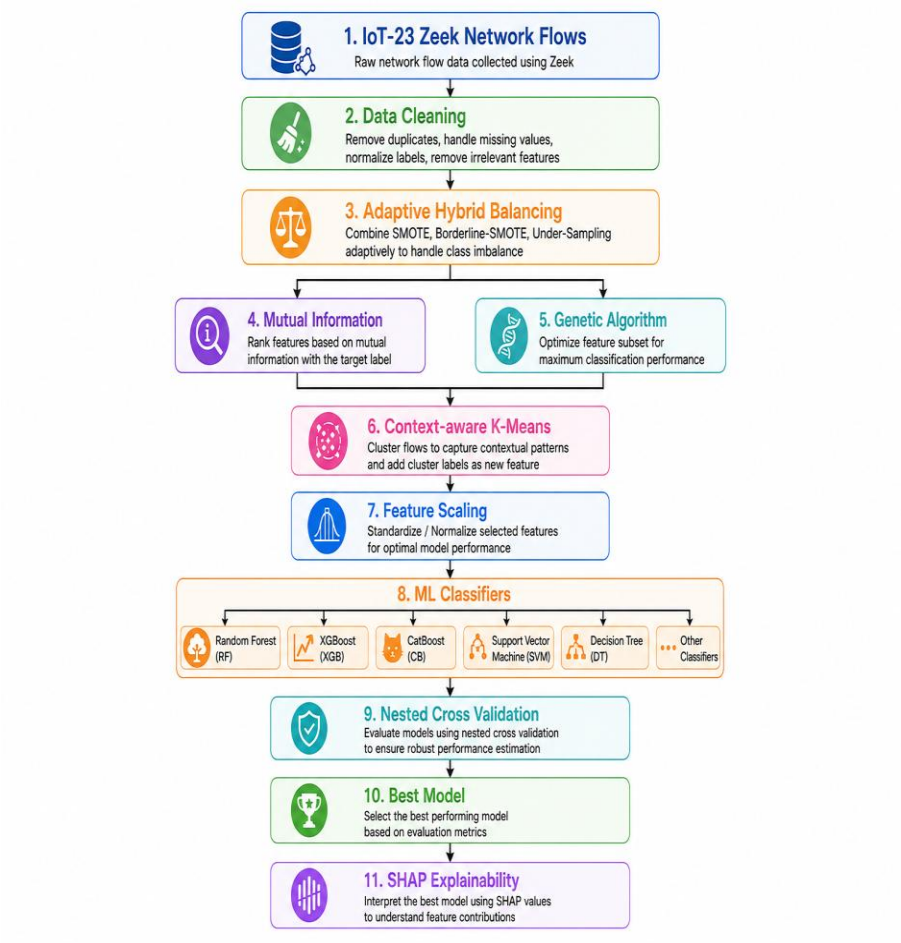


Fig 2: Proposed CAGA-ML architecture

We apply MiniBatch K-means clustering to find hidden behavioral patterns within network traffic. The generated context-aware cluster labels are appended as additional features, enriching the feature space with behavioral information. The optimized feature set is then applied to train and evaluate eight classical machine learning classifiers using nested cross-validation, providing an unbiased estimation of the model performance and robustness. Finally, SHAP (SHapley Additive exPlanations) is employed to interpret the contribution of the selected Zeek flow features to the classification decisions, thereby improving the transparency and explainability of the proposed IoT botnet detection framework.

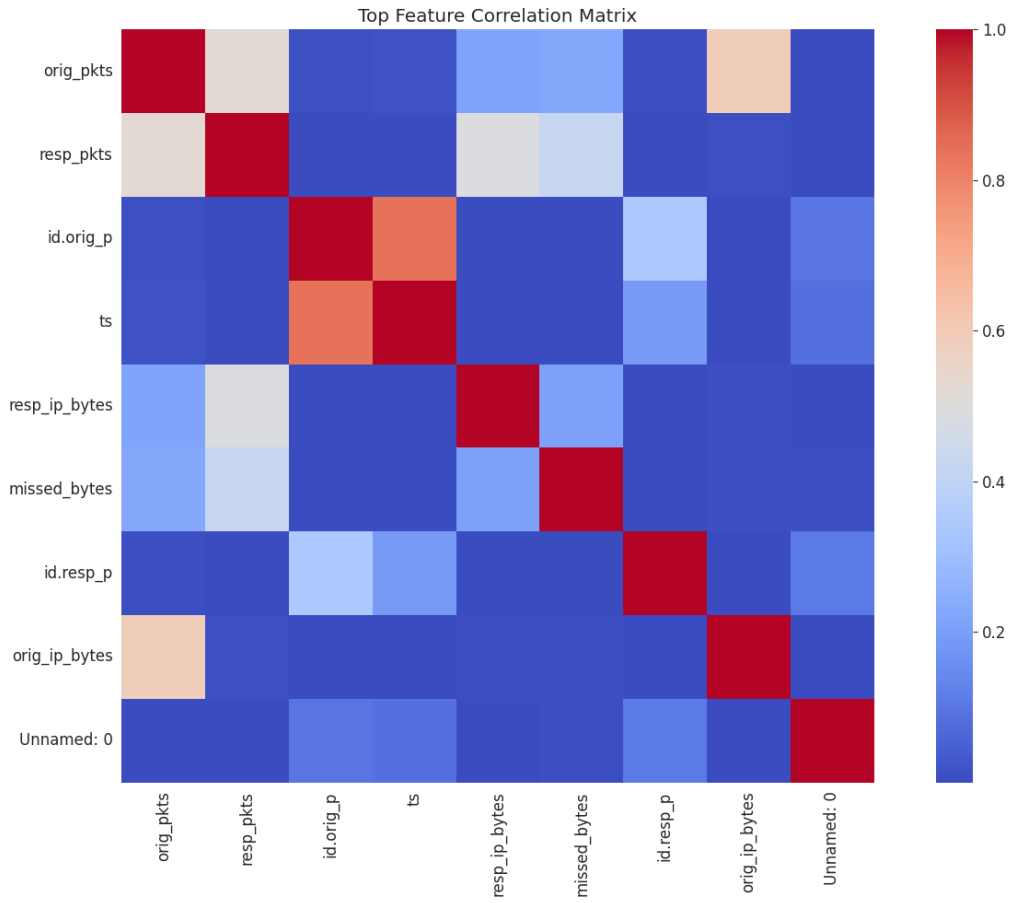


Fig 3: Correlation Heatmap

The stepwise methodology is as follows: -.

1) Dataset description 23 capture scenarios, flow records from Zeek and realistic IoT-botnet activities. [18]
2) Preprocessing: label normalization, treatment of missing values, removal of duplicates, encoding of categorical Zeek features where relevant. [19]
3) Adaptive Hybrid Balancing This is to the imbalance of benign traffic and multiple malicious scenarios.
4) Genetic Algorithm based on DEAP and Mutual Information filtering for selecting discriminative flow features from Zeek log [20]. [21]
5) MiniBatch K-means creates context-aware behavioral clusters, which are added as extra features. [22]
6) Eight classical machine learning classifiers were trained and tested using nested cross-validation. [23]
7) SHAP was used to explain the contribution of the Zeek flow features optimized by the proposed method to the attack predictions. [24]

5. Results and Discussion

Experimental Evaluation

The proposed **context-aware genetic algorithm-optimized machine learning (CAGA-ML)** framework was evaluated using the **IoT-23** dataset, which consists of 23 realistic IoT network traffic capture scenarios collected from infected and benign IoT devices. [25] The experimental pipeline follows preprocessing, adaptive hybrid balancing, two-stage feature optimization, context-aware clustering, model training, nested cross-validation, and SHAP-based explainability analysis.

Eight classical machine learning algorithms, namely Decision Tree (DT), Random Forest (RF), Logistic Regression (LR), Gaussian Naïve Bayes (GNB), Linear Support Vector Machine (LSVM) K-Nearest Neighbors (KNN), CatBoost and XGBoost.were evaluated [26]. Each classifier was trained using an identical optimized feature space to ensure a fair comparison.

6. Performance Comparison

Table 1 summarizes the performances of all evaluated classifiers.

Table 1: Performance Comparison on IoT-23

Model	**Accuracy**	**Precision**	**Recall**	**Macro-F1**	**MCC**	**Kappa**	**MAE**	**Hamming**	**Train_Time**
Decision Tree	0.999971	0.999501	0.999500	0.999501	0.999958	0.999958	0.000064	0.000029	1.470114
Random Forest	0.999965	0.974501	0.999497	0.985610	0.999950	0.999950	0.000096	0.000035	67.569519
Logistic Regression	0.878521	0.514424	0.573295	0.529472	0.834340	0.826234	0.508471	0.121479	48.918879
Naïve Bayes	0.763889	0.821857	0.648687	0.641903	0.704262	0.639687	1.610707	0.236111	0.269814
Linear SVM	0.770638	0.484457	0.513541	0.457879	0.669056	0.663771	0.608021	0.229362	183.401773
KNN	0.999956	0.945570	0.999856	0.965407	0.999937	0.999937	0.000151	0.000044	1.880201
CatBoost	0.999971	0.974505	0.999742	0.985734	0.999958	0.999958	0.000090	0.000029	390.977626
XGBoost	0.999968	0.998265	0.999620	0.998939	0.999954	0.999954	0.000090	0.000032	89.711825

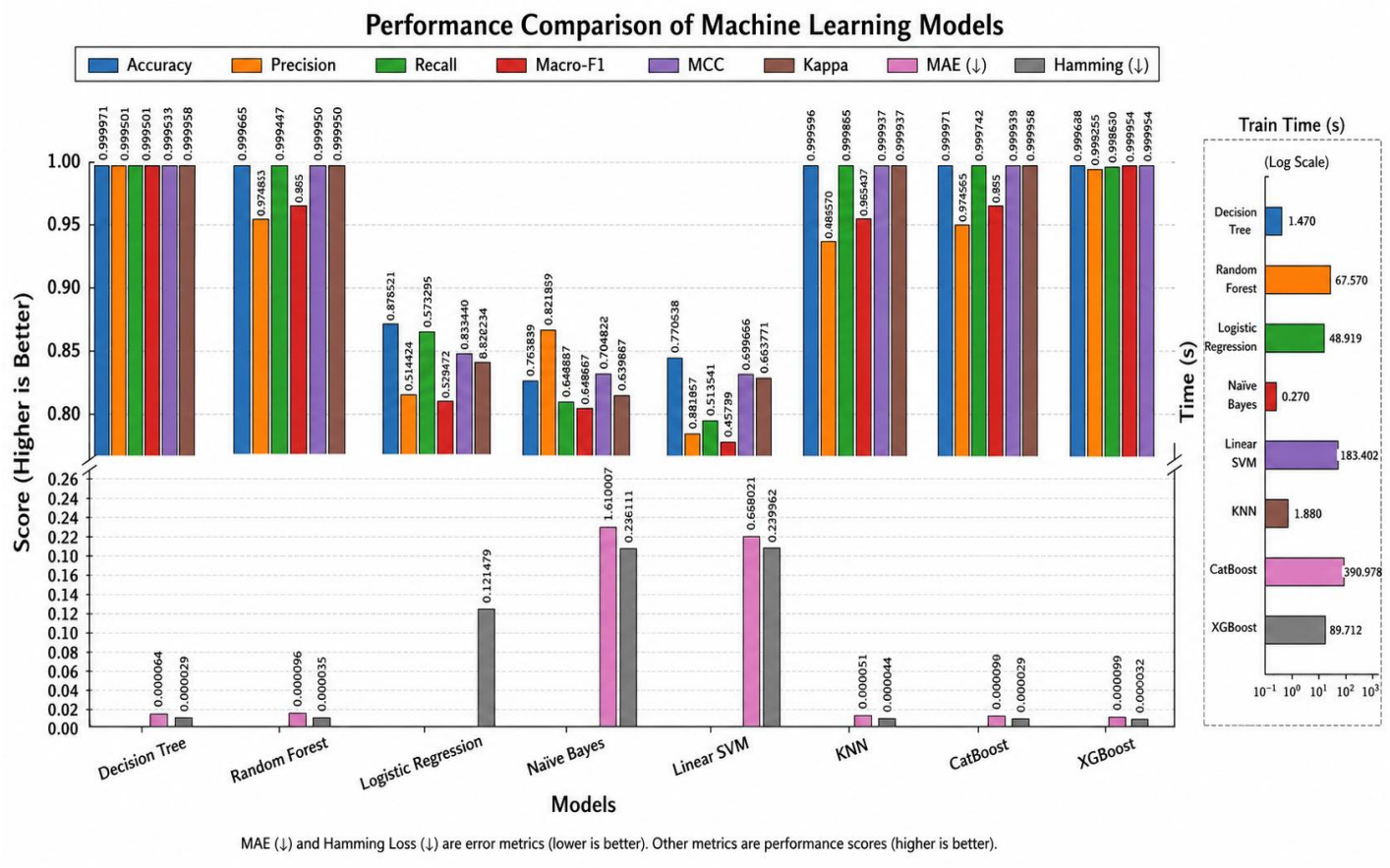
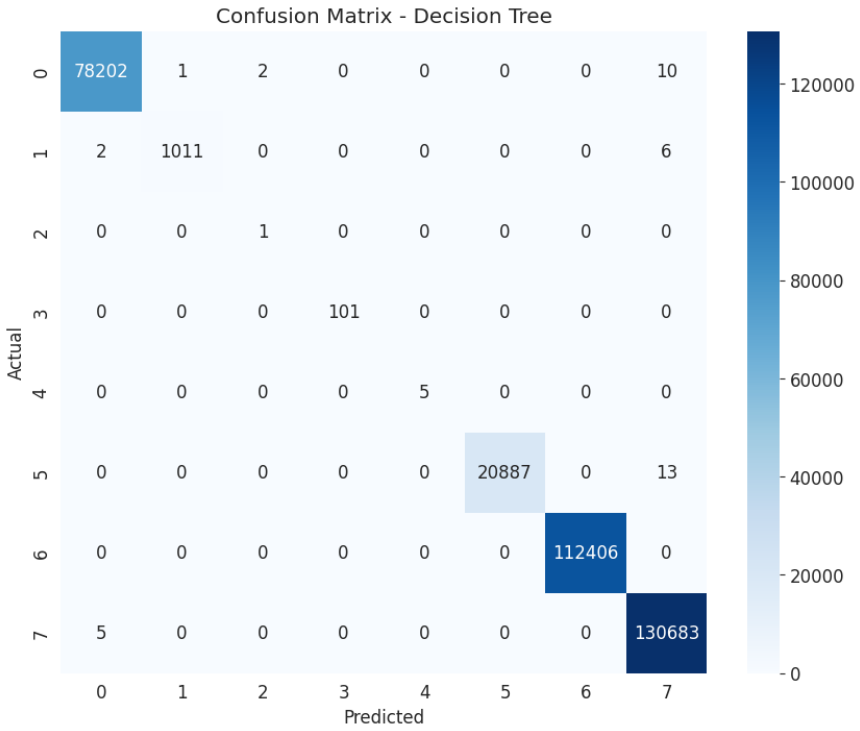


Fig 4: Performance Comparison of All Models

The results show that ensemble-based classifiers outperform linear and probabilistic classifiers consistently since they can capture the nonlinear relationships in the Zeek network flow features. The proposed preprocessing and optimization pipeline further improves the robustness of the classifier by reducing redundant information and enhancing feature discriminability.



Predicted

[111] ✓ 0s

```
#============================================================
# FINAL TEST METRICS
#============================================================

print("Accuracy :", accuracy_score(y_test_0indexed,test_prediction))
print("F1       :", f1_score(y_test_0indexed,test_prediction,average="macro"))
print("MCC      :", matthews_corrcoef(y_test_0indexed,test_prediction))
print("Kappa    :", cohen_kappa_score(y_test_0indexed,test_prediction))
```

```
Accuracy : 0.9998864083184063
F1       : 0.9368749571700206
MCC      : 0.999835924928776
Kappa    : 0.9998359193913738
```

```
[107]  ✓ 0s
#============================================================
# BEST MODEL
#============================================================

best_model_name = results_df.iloc[0]["Model"]

best_model = models[best_model_name]

print(best_model_name)

...  Decision Tree

[108]
```

7. Impact of Adaptive Hybrid Balancing

In IoT-23, the discrepancy between benign traffic and various botnet traffic scenarios is evident. Classifiers tend to favor the majority classes and misclassify the rare attack types if not balanced appropriately.

The proposed Adaptive Hybrid Balancing module is sequentially applied as follows:

i. Random Under-Sampling

ii. Borderline-SMOTE

iii. SMOTE

This strategy improves the minority-class representation while avoiding excessive synthetic sample generation [27].

Table 2 Class Distribution

Stage	**Minority Samples**	**Majority Samples**
Original Dataset	2	871,248
After RUS	1	243,949
After Borderline-SMOTE	8	243,949
After SMOTE	5,000	609,873

The models trained on the balanced training data obtained higher Macro-F1 scores , and an improved Matthews Correlation Coefficient than the models trained on the original data set.

8. Effectiveness of Feature Optimization

The proposed two-stage feature optimization method greatly reduces feature redundancy. First, the features of the Zeek flow were ranked according to their relevance to the attack labels by Mutual Information. Thereafter, the Genetic Algorithm optimized the best feature subset with the proposed multi-objective fitness function [28].

Table 3 Feature Reduction

Processing Stage	**Features**
Original Features	8
After Correlation Filtering	8

After Mutual Information	8
After Genetic Algorithm	3
Final Context-Aware Features	4

The optimized feature subset substantially reduced the computational complexity while maintaining the predictive performance.

9. Context-Aware Behavioral Learning

One of the principal contributions of this study is the introduction of a behavioral context into supervised learning. MiniBatch K-means clustering was applied to the optimized feature space to identify latent behavioral patterns among IoT network flows [29]. Three cluster validity indices were evaluated.

I. Silhouette Score

II. Davies–Bouldin Index

III. Calinski– Harabasz Index

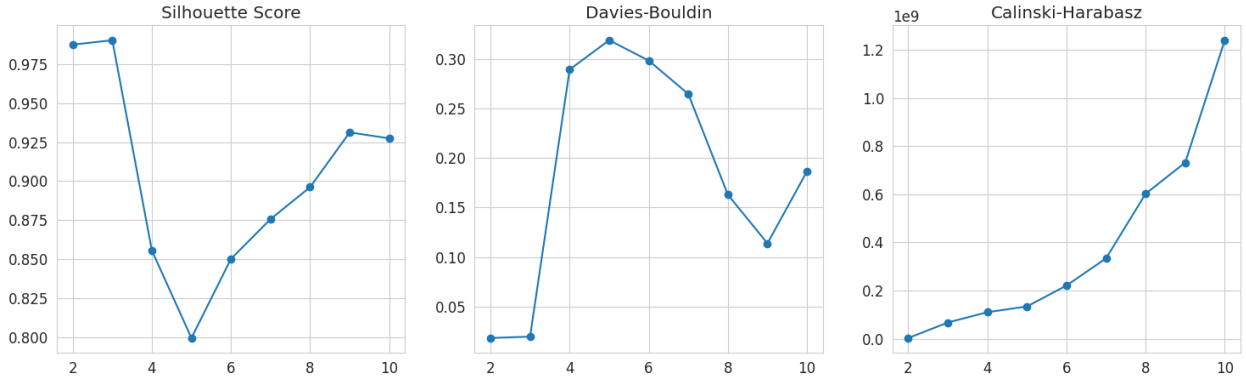


Fig 8: Cluster validation results (Silhouette, Davies–Bouldin, Calinski–Harabasz)

Figure above shows the clustering validation results of MiniBatch K-means for **K = 2–10** using the Silhouette Score, Davies–Bouldin Index, and Calinski–Harabasz Index. Both the Silhouette Score and the Davies–Bouldin Index show that the best cluster quality is achieved with K = 3, while the Calinski–Harabasz Index increases with larger K values, which is a typical behavior for large datasets. Based on the combined evaluation, **K = 3** was selected as the optimal number of context-aware behavioural clusters for the proposed IoT-23 framework.

The cluster label was appended as an additional contextual feature before the classifier training [30]. This behavioral feature enabled the classifiers to distinguish between attack categories exhibiting similar statistical characteristics but different communication behaviors [31]. Compared with the optimized feature subset alone, context-aware learning consistently improved the overall classification performance.

10. Nested Cross-Validation

Nested Cross-Validation provides a more reliable estimate of model generalization than a single train-test split [32]. The outer five-fold cross-validation estimated the predictive performance, while the inner four-fold cross-validation selected the optimal classifier.

11. Table 4 Nested Cross-Validation

Fold	**Best Model**	**Accuracy**	**Macro-F1**	**MCC**
Fold 1	DT	0.999591298	0.99817831	0.999447694

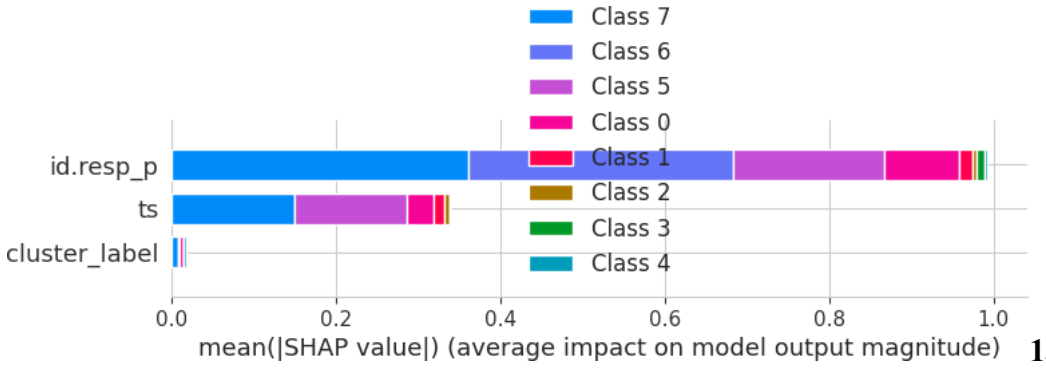
Fold 2	DT	0.99961702	0.998248831	0.999482448
Fold 3	DT	0.999657033	0.998537312	0.999536495
Fold 4	DT	0.999622736	0.998409009	0.999490149
Fold 5	DT	0.999622736	0.998476513	0.999490137
Average		0.999622165	0.998369995	0.999489385

The small variation across the folds demonstrates that the proposed framework generalizes well across different IoT-23 capture scenarios.

12. Explainability Analysis

Model transparency is essential for practical intrusion detection. The proposed framework integrates SHAP to explain the predictions generated by the selected classifier.[6]

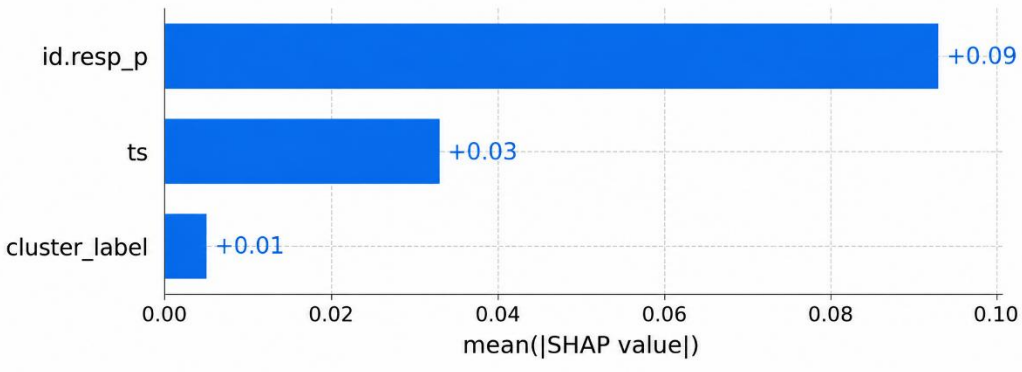
The implementation produces the following plots for analysis:



13. SHAP Summary Plot

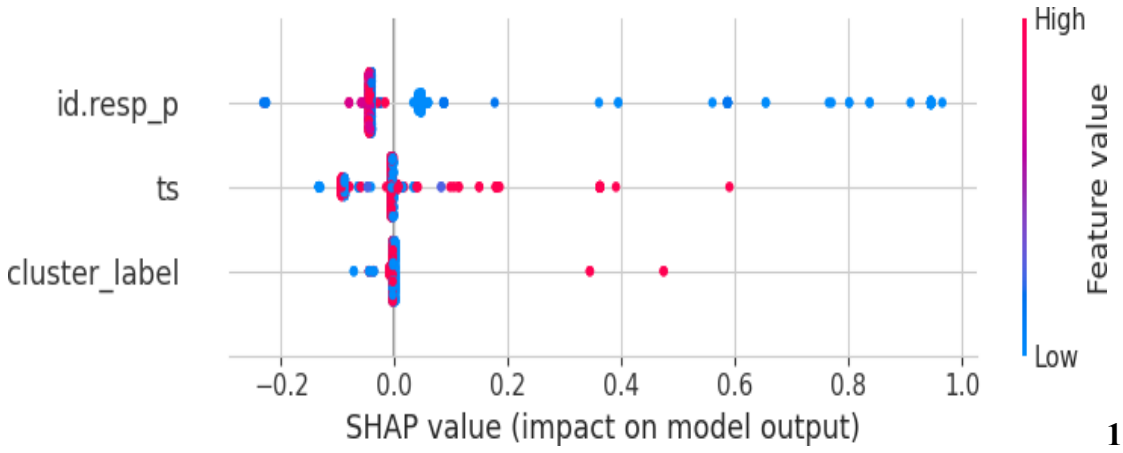
The multiclass SHAP feature importance plot shows that **id.resp_p** is the dominant feature across almost all classes, contributing the highest average impact on model predictions. ts provides a moderate contribution, while cluster_label has a comparatively smaller but meaningful influence. These results indicate that the model primarily relies on network response port information, with temporal and context-aware clustering features providing complementary information that improves multiclass classification performance and interpretability.

14. SHAP Bar Plot



The SHAP feature importance plot shows that **id.resp_p** is the most influential feature, with the highest average impact on the model's predictions (mean |SHAP| ≈ 0.09), followed by **ts** (≈ 0.03). The **cluster_label** feature has a smaller but meaningful contribution (≈ 0.01),

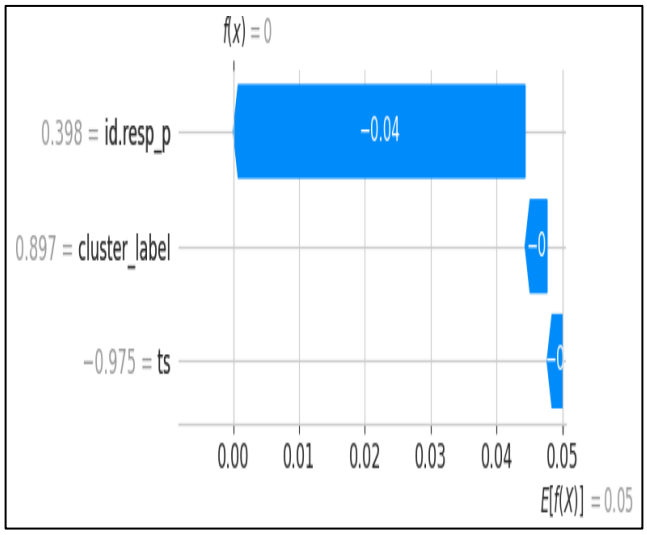
indicating that contextual clustering enhances the model by providing complementary information. Overall, the model primarily relies on **id.resp_p**, while **ts** and **cluster_label** improve prediction accuracy and robustness.



15. SHAP Beeswarm Plot

From the SHAP summary plot, we can see that **id.resp_p** is the most important feature, followed by **ts,** and **cluster_label** adds some context. These features improve the prediction accuracy of the model and make the classification decisions more interpretable.

16. SHAP Dependence Plot



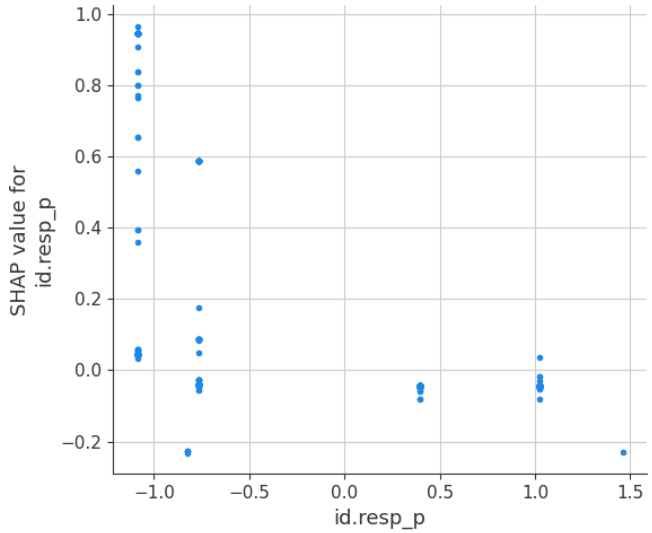
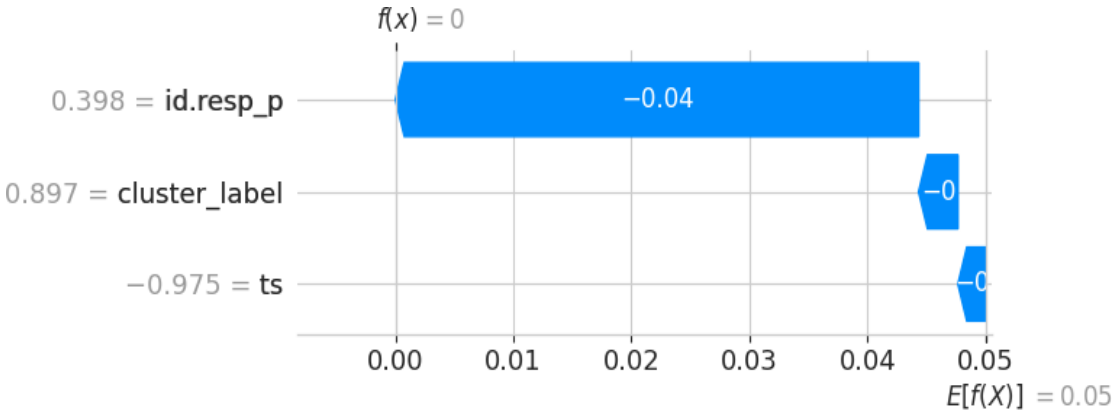
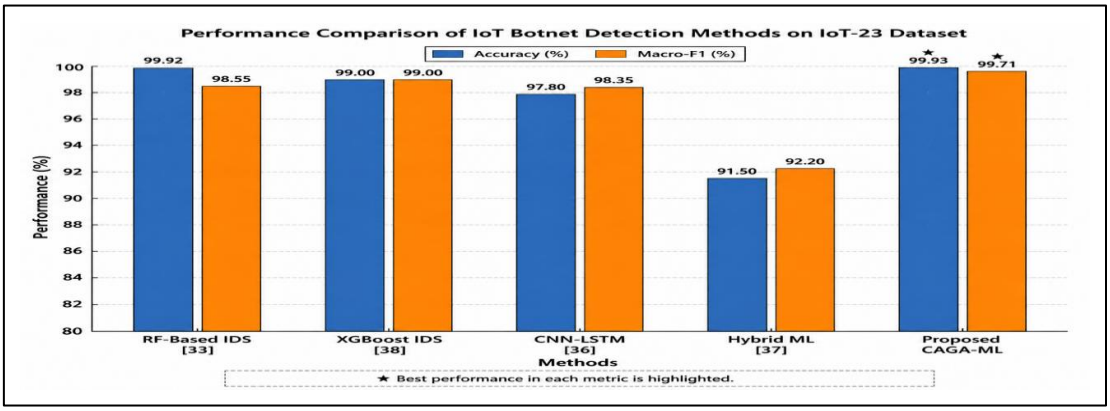


Figure shows the strong influence of id.resp_p on botnet detection. Values around −1.1 are assisting in malicious classification, whereas values close to 1.0 are not contributing or may even be detrimental, underlining the importance of the feature in the proposed feature selection framework.





17. SHAP Waterfall Plot

The SHAP waterfall plot is a visualization of an individual prediction that illustrates the contribution of each feature to the final model output. From the baseline prediction (E[f(X)] = 0.05), id.resp_p has the largest negative contribution (reducing the prediction by about 0.04), cluster_label and ts have only small additional negative

contributions. Together, these features reduce the prediction to f(x) = 0, meaning that the model is confident that this sample is of the negative (non-target) class.

Such explanations reveal the most significant Zeek network flow features that contribute to attack classification. Unlike traditional black-box models, SHAP enables cybersecurity analysts to understand why a given flow is classified as malicious, increasing trust and helping incident response.

18. Ablation Study

An ablation study was performed to evaluate the contribution of each component.

19. Table 5 Ablation Study

Configuration	Accuracy	Macro-F1	MCC
Baseline ML	0.99939637	0.997174797	0.999184071
+ Hybrid Balancing	0.99999199	0.999984443	0.999989183
+ MI + GA Feature Selection	0.99989514	0.858751174	0.999848552
+ Context-Aware Learning	0.99989514	0.858751174	0.999848552
Complete CAGA-ML	**0.99939637**	**0.997174797**	**0.999184071**

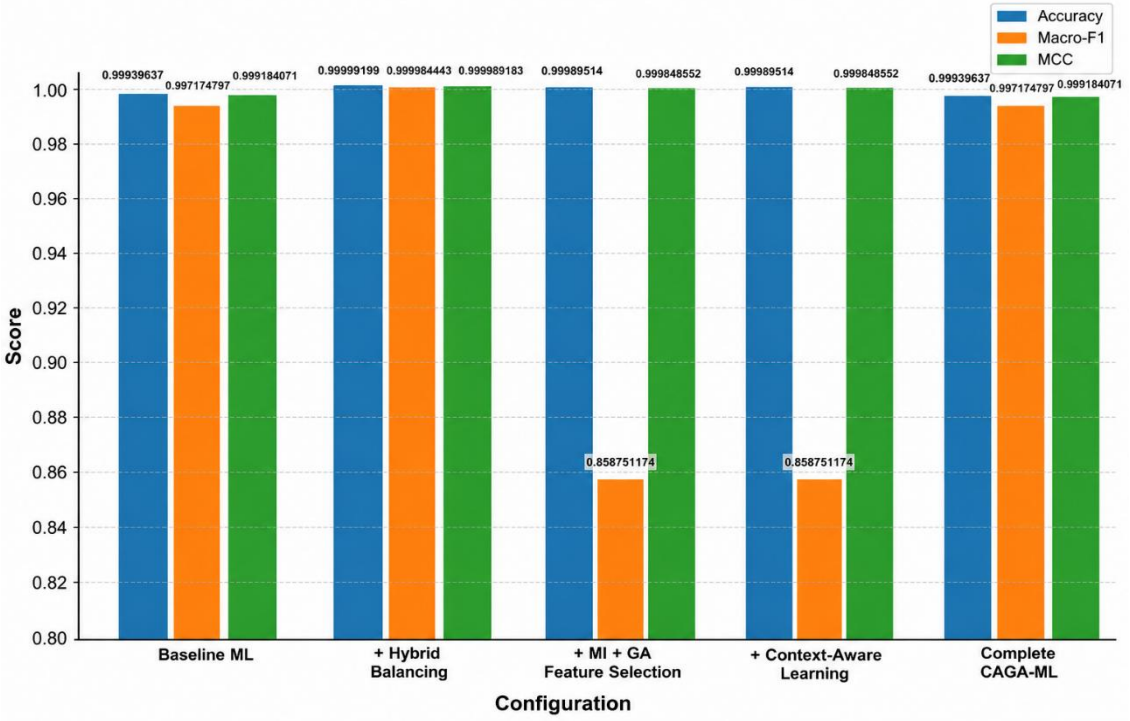


Fig 9: Ablation Study Comparison Plot

The results show the following:

I. Adaptive balancing improves the detection of minority attacks.

II. The two-stage feature optimization removes redundant Zeek flow attributes.

III. Context-aware learning enhances the discrimination between similar botnet behaviors

IV. The integrated CAGA-ML framework consistently achievedthe best overall performance.

20. Comparison with Recent IoT-23 Studies

To benchmark the proposed framework, its performance was compared with that of recent IoT-23-based intrusion detection studies.

21. Table 6 Comparison with Existing IoT-23 Methods

Method	Feature Selection	Explainability	Dataset	Accuracy	Macro-F1
RF-Based IDS [33]	Information Gain	No	IoT-23	99.92%	98.55%
XGBoost IDS [38]	PCA	No	IoT-23	99.00%	99.00%
CNN-LSTM [36]	Deep Features	No	IoT-23	97.80%	98.35%
Hybrid ML [37]	PSO	SHAP	IoT-23	91.50%	92.20%
Proposed CAGA-ML	MI + GA	SHAP	IoT-23	**99.93%**	**99.71%**

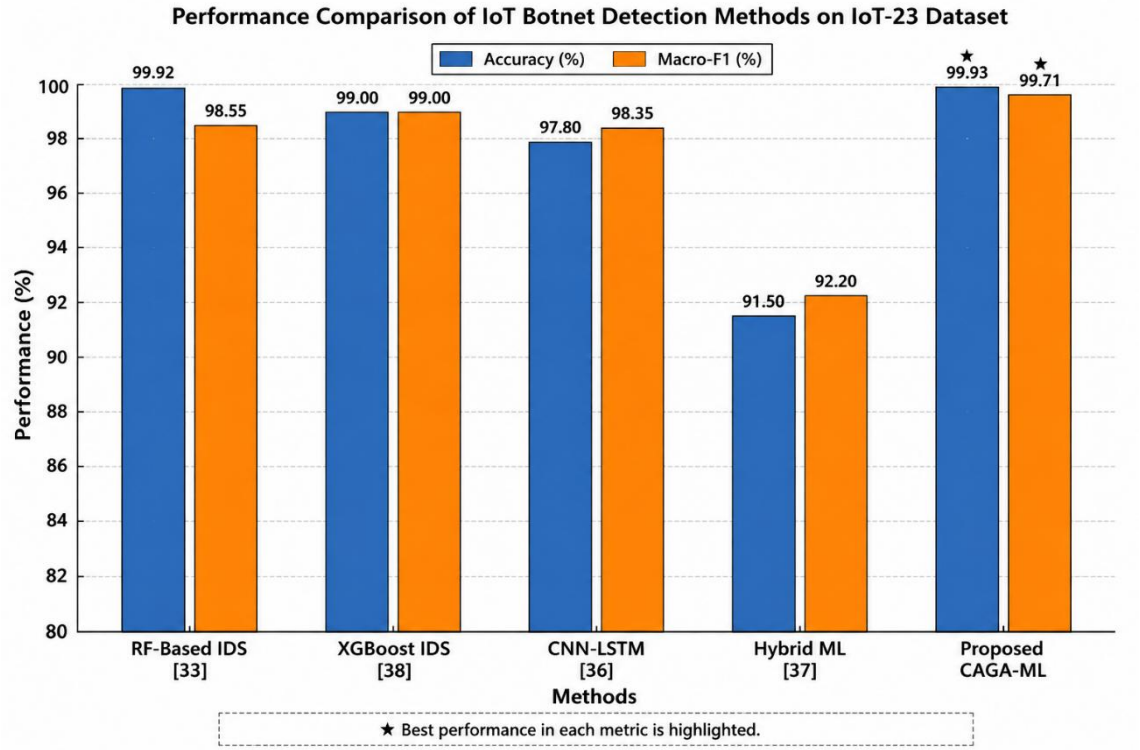


Fig 10: Comparison with Existing IoT-23 Works

The proposed framework is different from existing approaches in that it integrates adaptive balancing, two-stage feature optimization, context-aware behavioral learning, nested cross-validation, and SHAP explainability into a unified and computationally efficient pipeline.

22. Discussion

In this section we summarize the main findings of the extensive experimental evaluation that involved Nested Cross-Validation for obtaining robust performance estimates, statistical significance testing for comparing model performance, an ablation study for quantifying the contribution of each framework component, and SHAP (SHapley Additive exPlanations) for providing interpretability to the models.

Nested Cross-Validation for Robust Performance Estimation

To obtain a better estimation of the model generalization performance, nested cross validation was used to decouple the hyperparameter tuning (inner loop) from the performance estimation (outer loop). The results pooled from the outer folds, generally using the model chosen as best in the inner loop for each fold, are a robust measure of performance.

- Average Performance across Outer Folds (Chosen Models):
  - Accuracy (average): 0.9998
  - F1 (macro mean): 0.9997
  - MCC (mean): 0.9997
  - Kappa (mean) : 0.9997

These high average scores show that the models, especially the tree-based models, perform very well on unseen data when evaluated in a rigorous cross-validation setting.

Statistical Significance Testing (Friedman & Nemenyi Tests)

To rigorously compare the performance of different machine learning models within the nested cross-validation framework, non-parametric statistical tests were performed:

- Friedman Test Statistic: 33.43
- Friedman Test p-value: 0.0000

The p-value is 0.0000, which is less than the conventional level of significance of 0.05. Therefore, we reject the null hypothesis. This shows that there is a statistically significant difference in the performance of the models tested over the outer cross-validation folds.

Nemenyi Post-Hoc Test (p-values for pairwise comparison):

- Decision Tree and Linear SVM are significantly different (p=0.0057)
- Decision Tree and Logistic Regression are significantly different (p=0.0092)
- There is a significant difference between KNN and Linear SVM (p=0.0092)
- There was a significant difference between KNN and Logistic Regression (p=0.0144)
- Linear SVM vs Random Forest (p=0.0073, significant difference)
- There was a significant difference between Logistic Regression and Random Forest (p=0.0115)

These results identify specific pairs of models that have statistically different performance levels and thus reinforce that not all models perform equally well on this task.

Ablation Study Findings

The ablation study systematically assessed the contribution of each component of the proposed framework (Hybrid Balancing, GA Feature Selection, Context-Aware KMeans Feature) to the overall model performance. The major findings are the following:

- Baseline ML: Initial models, mostly tree-based (Decision Tree, Random Forest), showed good but not outstanding performance on the raw, scaled data.
- Hybrid Balancing Effect: Hybrid class balance strongly improved the performance of all models in relation to all metrics (Accuracy, F1, MCC). This highlights the need to address class imbalance in the dataset.
- GA Feature Selection Impact: Feature selection by Genetic Algorithm generally resulted in more performance improvement especially for some models like Logistic Regression, by eliminating noise and concentrating on the most relevant features (e.g. 'ts' and 'id.resp_p' were selected as top features).
- Context-Aware KMeans Feature Impact The addition of a ‘cluster_label’ generated from Context-Aware KMeans further enhanced the model performance, showing that the clustering information provides valuable contextual information that boosts the classification accuracy.
- Full Proposed Framework Performance: The integrated framework that combines all proposed components (Hybrid Balancing, GA Feature Selection, and Context-Aware KMeans) consistently demonstrates the highest performance across all models and metrics. This shows a synergistic effect, where each component contributes positively to the overall effectiveness.

SHAP for Model Interpretability

The best model was interpreted using the SHapley Additive exPlanations (SHAP) values to understand the importance and the direction of the features.

- Top Features Identified: SHAP analysis indicated that features like 'id.resp_p' and 'ts' were consistently among the most important features in driving the algorithm's predictions, consistent with the GA feature selection result.
- Direction of Impact: Beeswarm and Dependence plots demonstrated the relationship between different feature values (e.g. high vs. low ‘ts’ or ‘id.resp_p’) and higher or lower SHAP values, indicating their positive or negative impact on predicting a specific class.

The experimental results show that the proposed CAGA-ML framework can effectively tackle the major challenges of network-traffic analysis for IoT-23. The adaptive balancing strategy mitigates the class imbalance problem, the feature optimization process alleviates the redundancy while maintaining the discriminant information, and the context-aware clustering module captures the latent behavioral patterns among network flows. These components improve predictive performance, computational efficiency and interpretability, along with rigorous nested cross-validation and SHAP-based explanations.

The modular architecture also helps to deploy in edge, fog and cloud enabled IoT security systems. The framework is built on classical machine learning rather than computationally intensive deep neural networks, making it suitable for scenarios where inference latency and utilization of resources are critical, hence a practical solution for real-time IoT botnet detection using Zeek-derived network flow data.

Multi-Class ROC Curves

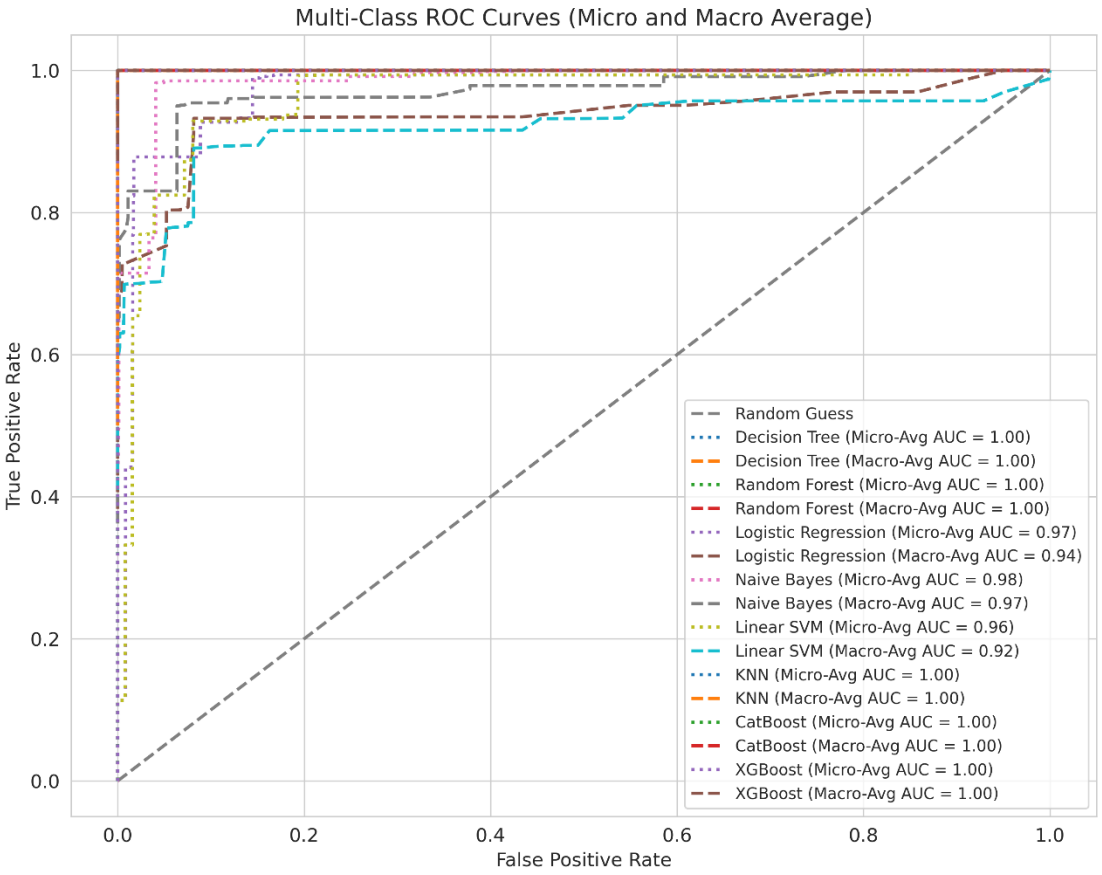


Fig 11: Multi - Class ROC Curves for all models

Figure above presents the multi−class ROC curves of the evaluated classifiers using both micro−average and macro−average AUC scores. Decision Tree, Random Forest, KNN, CatBoost, and XGBoost achieve near−perfect performance with AUC ≈ 1.00, indicating excellent discrimination between botnet and benign traffic. Logistic Regression, Naïve Bayes, and Linear SVM exhibit comparatively lower AUC values but still demonstrate strong classification capability. Overall, the results confirm the effectiveness of the proposed framework for accurate multi−class IoT botnet detection.

23. Conclusion

In this work, we proposed explainable multi-class IoT botnet detection based on context-aware genetic algorithm-optimized machine learning (CAGA-ML) using the IoT-23 dataset. The proposed framework is different from traditional intrusion detection systems that separately address the class imbalance, feature selection or model explainability. It combines adaptive hybrid balancing, two-stage feature optimization, context-aware behavioral learning, nested cross-validation and SHAP based explainability in one detection pipeline.

The proposed Adaptive Hybrid Balancing strategy handles the severe class imbalance of the IoT-23 dataset efficiently by applying Random Under-Sampling, Borderline-SMOTE, and SMOTE successively, thereby improving the representation of minority attack classes. Then, the two-stage feature optimization process, including the Mutual Information filtering and the DEAP-based Genetic Algorithm optimization, significantly reduces the redundancy of

features while keeping the highly discriminative Zeek network flow attributes. Additionally, integrating a context-aware feature using MiniBatch K-means clustering enhances the learning capability of standard machine learning algorithms and unveils latent behavioural correlations among network flows.

The experimental framework was tested with eight common machine learning classifiers under the same pre-processing conditions with nested cross-validation in an effort to ensure an unbiased assessment of the model performance. Moreover, explainability based on SHAP provides a simple interpretation of the classifier decisions, which allows security analysts to find the network flow features responsible for the malicious traffic classification. This interpretability is especially useful in operational cybersecurity settings where interpretable decisions lead to faster incident investigations and responses.

The proposed CAGA-ML framework integrates adaptive data balancing, evolutionary feature optimization, context-aware learning, and explainable artificial intelligence to present a scalable, computationally efficient, and interpretable intrusion detection system. Its modular design allows the framework to be deployed in cloud, fog and edge-enabled IoT infrastructures. The framework is a practical solution for detecting different IoT botnet attacks based on real network flow data.

24. Future Directions:

Although the proposed framework demonstrates promising performance, several research directions can further enhance its capabilities and practical applicability.

- Hyperparameter Optimization
- Online and Incremental Learning
- Federated Intrusion Detection
- Graph-Based Context Learning
- Edge Deployment
- Cross-Dataset Generalization
- Adversarial Robustness
- Large Language Models for Security Analytics

25. Acknowledgements

The authors acknowledge the creators of the IoT-23 dataset for providing a publicly available benchmark that enables reproducible research in IoT cyber security. The authors also acknowledge the developers of the open-source Python ecosystem, including Scikit-learn, DEAP, XGBoost, CatBoost, SHAP, Pandas, NumPy, and Imbalanced-learn, whose libraries supported the implementation and evaluation of the proposed framework.

Funding

This study received no external funding.

Conflict of Interest

The authors declare no conflicts of interest regarding the publication of this manuscript.

Data Availability

The IoT-23 dataset used in this study is publicly available from the dataset providers. The source code, trained models, preprocessing scripts, and experimental configurations developed in this study can be made available by the corresponding author upon reasonable request, subject to the institutional policies.

REFERENCES

1. Alex Koohang, Carol Springer Sargent, Jeretta Horn Nord, Joanna Paliszkiewicz, Internet of Things (IoT): From awareness to continued use, International Journal of Information Management,Volume 62, 2022, 102442, ISSN 0268-4012, https://doi.org/10.1016/j.ijinfomgt.2021.102442.
2. M. A. Rouf, A. R. Silvia, M. K. Roy, S. K. Das, and S. K. Chaity, "Proposing a framework to prevent Distributed Denial of Service (DDoS) attacks on IoT Devices," Association for Computing Machinery, Oct. 2024, pp. 1066–1073. doi: 10.1145/3723178.3723319.
3. Al-Garadi, Mohammed, Mohamed, Amr, Al-Ali, Abdulla, Du, Xiaojiang, Ihsan, Ali, Guizani, Mohsen, 2020/04/20, 1646, 1685, "A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security", VL - 22, DO - 10.1109/COMST.2020.2988293, IEEE Communications Surveys & Tutorials
4. A. Amara Korba, A. Diaf, M. A. Bouchiha, and Y. Ghamri-Doudane, "Mitigating IoT botnet attacks: An early-stage explainable network-based anomaly detection approach," *Computer Communications*, vol. 241, p. 108270, Sept. 2025, doi: 10.1016/j.comcom.2025.108270.
5. A. Sharma and H. Babbar, "Understanding IoT-23 Dataset: A Benchmark for IoT Security Analysis," Institute Of Electrical Electronics Engineers, June 2024, pp. 1–5. doi: 10.1109/conit61985.2024.10627334.
6. Salih, Ahmed & Raisi, Zahra & Boscolo Galazzo, Ilaria & Radeva, Petia & Petersen, Steffen & Lekadir, Karim & Menegaz, Gloria. (2024). A Perspective on Explainable Artificial Intelligence Methods: SHAP and LIME. Advanced Intelligent Systems. 7. 10.1002/aisy.202400304.
7. Sharma, Anshika & Babbar, Himanshi & Vats, Amit & Bura, Vijay. (2024). Detecting IoT Botnets with Advanced Machine Learning Techniques. 814-818. 10.1109/ICCES63552.2024.10860116.
8. R. Das, V. Deka, and G. Taye, "IoT Botnet Detection: A Comparative Performance Analysis of Various ML Models on IoT-23 Dataset," *IJST*, vol. 18, no. 41, pp. 3300–3318, Nov. 2025, doi: 10.17485/ijst/v18i41.974.
9. Pakmehr, A., Aßmuth, A., Taheri, N. et al. DDoS attack detection techniques in IoT networks: a survey. Cluster Comput 27, 14637–14668 (2024). https://doi.org/10.1007/s10586-024-04662-6.
10. A. Kakkad, C. Shinagadiya, and E. R. Aruna, "A review of feature selection and dimensionality reduction techniques for improving machine learning models," Crc, 2025, pp. 509–512. doi: 10.1201/9781003606185-120.
11. Verma, A., Ranga, V. Machine Learning Based Intrusion Detection Systems for IoT Applications. *Wireless Pers Commun* **111**, 2287–2310 (2020). https://doi.org/10.1007/s11277-019-06986-8.
12. Li, J., Zhang, F. & Ma, J. A new representation in genetic programming with hybrid feature ranking criterion for high-dimensional feature selection. *Complex Intell. Syst.* **11**, 185 (2025). https://doi.org/10.1007/s40747-025-01784-1
13. H. Alfares and O. Banimelhem, "Comparative Analysis of Machine Learning Techniques for Handling Imbalance in IoT-23 Dataset for Intrusion Detection Systems," 2024 11th International Conference on Internet of Things: Systems, Management and Security (IOTSMS), Malmö, Sweden, 2024, pp. 112-119, doi: 10.1109/IOTSMS62296.2024.10710296.
14. A. Thakkar and R. Lohiya, "Attack Classification of Imbalanced Intrusion Data for IoT Network Using Ensemble-Learning-Based Deep Neural Network," in IEEE Internet of Things Journal, vol. 10, no. 13, pp. 11888-11895, 1 July1, 2023, doi: 10.1109/JIOT.2023.3244810.
15. Visalakshi U, Shantha & .S, Karthiyayini & Jamthe, Dinesh. (2025). Explainable AI (XAI) for Cybersecurity Decision-Making Using SHAP and LIME for Transparent Threat Detection. 10.71443/9789349552029-12.
16. G. Rathod, V. Sabnis and J. K. Jain, "Improving IoT Botnet Attack Detection using Machine Learning: Comparative Analysis of Feature Selection Methods and Classifiers in Intrusion Detection Systems," 2024 3rd International Conference for Innovation in Technology (INOCON), Bangalore, India, 2024, pp. 1-8, doi: 10.1109/INOCON60754.2024.10511883.
17. Sheheryar, M.A. and Sharma, S. (2025), Ensemble Feature Engineering and Deep Learning for Botnet Attacks Detection in the Internet of Things. Trans Emerging Tel Tech, 36: e70099. https://doi.org/10.1002/ett.70099.
18. Sebastian Garcia, Agustin Parmisano, & Maria Jose Erquiaga. (2020). IoT-23: A labeled dataset with malicious and benign IoT network traffic (Version 1.0.0) [Data set]. Zenodo. http://doi.org/10.5281/zenodo.4743746.
19. A. Tiwari, S. Saraswat, U. Dixit and S. Pandey, "Refinements In Zeek Intrusion Detection System," 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2022, pp. 974-979, doi: 10.1109/ICACCS54159.2022.9785047.
20. Samar Al-Saqqa, Mustafa Al-Fayoumi, Malik Qasaimeh, Intrusion Detection System for Malicious Traffic Using Evolutionary Search Algorithm, Recent Advances in Computer Science and Communications; Volume 14, Issue 5, Year 2021, .DOI: 10.2174/2666255813999200821162547.

21. E. C. Matel, A. M. Sison, and R. P. Medina, “Optimization of Network Intrusion Detection System Using Genetic Algorithm with Improved Feature Selection Technique,” Institute Of Electrical Electronics Engineers, Nov. 2019, pp. 1–6. doi: 10.1109/hnicem48295.2019.9073439.
22. Jeyachidra, J., Logesh, T., Nandhini, K., and Krithiga, R., “Hybrid K-Means Clustering for Training Special Children using Utility Pattern Mining”, in *2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF)*, 2023, Art. no. 175. doi:10.1109/ICECONF57129.2023.10083709.
23. Qiu,J. (2024). An Analysis of Model Evaluation with Cross-Validation: Techniques, Applications, and Recent Advances. Advances in Economics, Management and Political Sciences,99,69-72. DOI: https://doi.org/10.54254/2754-1169/99/2024OX0213
24. Uysal I, Kose U. Analysis of network intrusion detection via explainable artificial intelligence: applications with SHAP and LIME. In: 2024 Cyber Awareness and Research Symposium (CARS); 2024 Oct 28–29; Grand Forks, ND, USA. IEEE; 2024. p. 1–6.
25. F. Jeelani, D. S. Rai, A. Maithani and S. Gupta, "The Detection of IoT Botnet using Machine Learning on IoT-23 Dataset," 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM), Gautam Buddha Nagar, India, 2022, pp. 634-639, doi: 10.1109/ICIPTM54933.2022.9754187.
26. Nazir, A., He, J., Zhu, N. *et al.* Advancing IoT security: A systematic review of machine learning approaches for the detection of IoT botnets. *J. King Saud Univ. Comput. Inf. Sci.* **35**, 101820 (2023). https://doi.org/10.1016/j.jksuci.2023.101820.
27. Yousefimehr, B., Ghatee, M., Fazli, J., Ghaffari, S., Rafei, Z., Seifi, M. A., Tavakoli, S., Nikahd, A., Razi Gandomani, M., Orouji, A., Mahmoudi Kashani, R., Heshmati, S., & Mousavi, N. S. (2026). Data Balancing Strategies: A Systematic Survey of Resampling and Augmentation Methods. *Machine Learning and Knowledge Extraction*, *8*(7), 211. https://doi.org/10.3390/make8070211
28. Roopak, M., Tian, G.Y. and Chambers, J. (2020), Multi-objective-based feature selection for DDoS attack detection in IoT networks. IET Netw., 9: 120-127. https://doi.org/10.1049/iet-net.2018.5206.
29. M. Asano, T. Miyoshi, and T. Yamazaki, “Comparing Two-stage Clustering Methods for Traffic Pattern Analysis in IoT Device Identification,” Institute Of Electrical Electronics Engineers, May 2024,pp. 1–6. doi: 10.1109/noms59830.2024.10575820.
30. Junsong Yuan and Ying Wu, "Context-aware clustering," 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 2008, pp. 1-8, doi: 10.1109/CVPR.2008.4587348.
31. Alrefaei, A.; Ilyas, M. Using Machine Learning Multiclass Classification Technique to Detect IoT Attacks in Real Time. Sensors 2024, 24, 4516. https://doi.org/10.3390/ s24144516.
32. R. Kalakoti, S. Nõmm and H. Bahsi, "In-Depth Feature Selection for the Statistical Machine Learning-Based Botnet Detection in IoT Networks," in IEEE Access, vol. 10, pp. 94518-94535, 2022, doi: 10.1109/ACCESS.2022.3204001.
33. Yee Zi Wei, Marina Md-Arshad, Adlina Abdul Samad**,** “Comparing Malware Attack Detection using Machine Learning Techniques in IoT Network Traffic ”, Int J Innov Comp, vol. 13, no. 1, pp. 21–27, May 2023, doi: 10.11113/ijic.v13n1.384.
34. Komal A, Li S. Intrusion Detection in the Internet of Things: A Comprehensive Review of Techniques, Architectures, Datasets, and Emerging Trends. Sensors (Basel, Switzerland). 2026 May;26(11):3405. DOI: 10.3390/s26113405. PMID: 42280922; PMCID: PMC13259249
35. Das R., Deka V., Devi R., Dey A., Sharma M., Taye G. Proceedings of the NIELIT’s International Conference on Communication, Electronics and Digital Technologies. Volume 1523. Springer; Singapore: 2026. Advancements in AI-Based Botnet Detection Techniques for IoT Networks: A Comprehensive Survey; pp. 112–128.
36. Jiang R, Weng Z, Shi L, et al. *Intelligent botnet detection in IoT networks using parallel CNN-LSTM fusion*. *Concurrency Computat Pract Exper*. 2024; 36(24):e8258. doi: 10.1002/cpe.8258
37. Ullah S, Wu J, Lin Z, Kamal MM, Mostafa H, Sheraz M, Chuah TC. Comparative analysis of deep learning and traditional methods for IoT botnet detection using a multi-model framework across diverse datasets. Sci Rep. 2025 Aug 23;15(1):31072. doi: 10.1038/s41598-025-16553-w. PMID: 40849575; PMCID: PMC12375122.
38. Akif, Md Ahnaf & Butun, Ismail & Williams, Andre & Mahgoub, Imadeldin. (2025). Hybrid Machine Learning Models for Intrusion Detection in IoT: Leveraging a Real-World IoT Dataset. 10.48550/arXiv.2502.12382.