From IoT Vulnerabilities to Intrusion Detection: An Explainable Vulnerability-Aware Machine Learning Framework for Smart Home IoT Security

Huda Aldawghan, Mounir Frikha
Department of Computer Networks and Communications,
College of Computer Sciences and Information Technology, King Faisal University, Al-Ahsa, 31982 Saudi Arabia

***Abstract*—The swift deployment of IoT-based smart home appliances has increased the attack surface for the smart environment and exposed it to attacks like botnet command-and-control communications, brute force attacks, denial-of-service attacks, and web-based attacks. Even though the Intrusion Detection Systems (IDSs) that use Machine Learning (ML) algorithms achieve a very high detection rate, most existing solutions focus on predictive performance but lack the ability to link detected attacks to the corresponding vulnerabilities in the Internet of Things (IoT). In this paper, an interpretable vulnerability-aware ML-based approach is presented to solve this problem through the integration of vulnerability classes of IoT, attack classes, network flow attributes, and ML features into one interpretation model. The proposed method uses leakage-aware preprocessing, addressing class imbalance, and compares Random Forest, XGBoost, and soft voting ensemble ML techniques using the CSE-CIC-IDS2018 dataset. Experimental outcomes indicate that XGBoost outperforms the other approaches in terms of performance, with an accuracy of 98.22%, precision of 99.69%, F1-score of 95.36%, ROC-AUC of 99.07%, and only 760 false alarms, which is approximately 19× lower number of false positives compared to Random Forest while keeping a similar level of detection efficiency. In addition to numeric assessment of the approach performance, the suggested model provides the vulnerability-oriented interpretation module that establishes mapping between prominent network flow attributes and possible IoT vulnerability states and attacks. Therefore, the integration of an interpretable vulnerability reasoning component into a high-performing tree-based machine learning algorithm proves to be effective for smart home IoT intrusion detection.**

***Keywords*—IoT security; intrusion detection system; machine learning; XGBoost; random forest; vulnerability-aware security; botnet detection; smart home; explainable AI; network-flow analysis**

I. Introduction

A. Background

With the proliferation of IoT devices in smart homes and edge networks, the management of digital systems has been radically changed [9], [24] , [27] , [28]. IoT and smart-home devices are vulnerable to well-known issues, namely default passwords, unsecured administration interfaces, legacy communication protocols, outdated firmware, and a lack of web-layer security [1] , [9], [24]. These vulnerabilities could make connected environments vulnerable to adversaries that could use some of these devices to recruit them for a botnet, make brute force credential attacks, establish a persistent C2 network, or launch DoS or DDoS attacks. The traditional signature-based IDSs are based on the predefined signature and cannot be generalized to new attacks [1], [29]. The three main IDS paradigms identified as signature-based, anomaly-based, and ML-based have significant differences in their way of processing network traffic before making a detection decision as shown in Fig. 1. In this study, ML-based approaches were used to learn patterns of behaviors directly from labelled training data without manually updating the rules [1], [25], [30], [33].

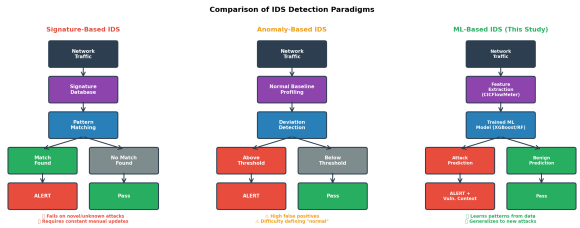

Fig. 1. Comparison of the three IDS detection paradigms.

As shown in Fig. 1 Comparison of the three IDS detection paradigms: signature-based (fails on novel attacks), anomaly-based (high false positives), and ML-based (this study learns generalizable patterns). Original diagram created for this paper.

Tree-based ensemble models, in particular, Random Forest [4] and XGBoost [5] have proven to be highly successful on structured, tabular network-flow data [1], [6], [31]. Breiman's Random Forest [4] showed decorrelated tree ensembles to be a reliable method for reducing variance while also maintaining accuracy. The efficient split-finding regularized gradient boosting method XGBoost was proposed by Chen and Guestrin [5].

B. Problem Statement and Motivation

One of the main problems with the current research on IDS is that most published research reports classification metrics but does not explicitly relate model output to the vulnerability conditions which allow an attack to be detected [3] [32]. But a benign/malicious classification isn't enough, security operators need proof that an alert is the result of brute-force credential probing, C2 timing anomalies, flooding behavior or another attack-related pattern [3]. The conceptual backbone of the vulnerability-aware interpretation developed in Section 7 is shown in Fig. 2 as the end-to-end chain from IoT vulnerability to attack behavior to network-flow signature

Fig. 2. Vulnerability-to-detection chain.

to ML feature detection. This paper fills the identified gap by mapping the most prominent learned features to plausible real-world vulnerability conditions, which align with the vision of explainable cybersecurity proposed by Sarker [3] and the existing XAI literature [7], [10], [12], [33].

As shown in Fig. 2 Vulnerability-to-detection chain: from IoT root cause (weak credentials, exposed ports, outdated firmware, weak web protection) through attack manifestation, to network-flow signature, to the dominant ML features identified by the Random Forest model. Original diagram created for this paper.

For this research, “vulnerability-aware" pertains to the security interpretation layer introduced following intrusion prediction and feature attribution layers, and does not mean that machine learning algorithms can predict software vulnerabilities. Specifically, the proposed approach involves three main steps: 1) intrusion prediction, where the classifier trained for this purpose determines whether or not traffic flow belongs to a benign or malicious group; 2) feature attribution, which evaluates the impact of traffic behavior features on the model behavior; and 3) security interpretation based on IoT vulnerabilities, where traffic behaviors contributing to model decisions are translated into the vulnerabilities of IoT systems using existing security literature. Thus, vulnerability translation is considered a subsequent interpretational step, not an additional classification model.

C. Objectives and Contributions

The main achievements of this paper are:

- Complete, reproducible end-to-end IDS workflow on CSE-CIC-IDS2018 [2] explicit leakage prevention and class imbalance handling.
- A direct comparative study of Random Forest [4], XGBoost [5] and RF+XGBoost soft voting ensemble [6] showing that XGBoost achieves the best results with precision of 0.9969 and 760 false positives (19 times fewer than RF).
- Original arrow diagrams (Fig. 1 , Fig. 2, Fig. 3, Fig. 4, Fig. 5 basing the ML methodology in well-established learning theory.
- A matrix of IoT vulnerabilities (Table II) to their corresponding network-flow signatures.
- An organized vulnerability-based interpretation layer (Table VII and Table XIV + Fig. 17, Fig. 2) that correlates dominant behavioral network flow patterns and their attacker manifestations with likely IoT vulnerabilities. This correlation provides security context to intrusion detection results in hindsight rather than predicting vulnerabilities.
- The actual implemented pipeline is the basis for a detailed ablation study (Table XV) and scalability analysis (Table XVI).
- A roadmap of future research (Table XVII) that is structured for directions.

D. Paper Organization

Section 2 introduces the theoretical background, including original arrow diagrams of RF, XGBoost, and ensemble architectures. Section 3 summarizes the related work. In Section 4, the classes of IoT vulnerabilities are characterized. The experimental setup is described in Section 5. The results are presented in Section 6. The explainability and vulnerability-aware discussion is provided in Section 7. The discussion is given in Section 8. At the end of Section 9, the Future Work Roadmap is provided.

II. Background And Theoretical Foundations

A. Machine Learning for Intrusion Detection

Intrusion detection based on ML assumes that malicious network traffic has statistically identifiable behavioral patterns that are different from benign network traffic [1], [25]. Supervised learning is used to train a classifier using labelled examples, and thus generalizing to new flows. CICFlowMeter [2] can generate 80 bidirectional flow features per connection such as inter-arrival times, packet lengths, number of bytes, numbers of flags, window sizes, etc., without DPI [13], [14], [30]. Fig. 1 compares and contrasts the ML-based approach with the signature-based and anomaly-based approaches, highlighting the benefits of the ML-based approach and its ability to overcome the disadvantages of both signature-based and anomaly-based approaches.

B. Random Forest Theoretical Basis

Random Forest (RF) [4] proposed by Breiman in 2001 is a method that grows many decision trees over bootstrap samples of the training data. At each split, each tree will make use of a random subset of features, which will decorrelate the trees and decrease ensemble variance. This is the RF architecture used in this study: 200 trees trained on bootstrap samples of the 5,327,625 training flows, with MDI feature importance extracted after training, as shown in Fig. 3. Both key theoretical properties and empirical performance analysis reveal that: 1) variance reduction can be achieved by simply averaging; 2) robustness to overfitting through bootstrap sampling and feature randomization; and 3) natural estimation of the importance of features by using mean decrease in impurity (MDI) [4], [31].

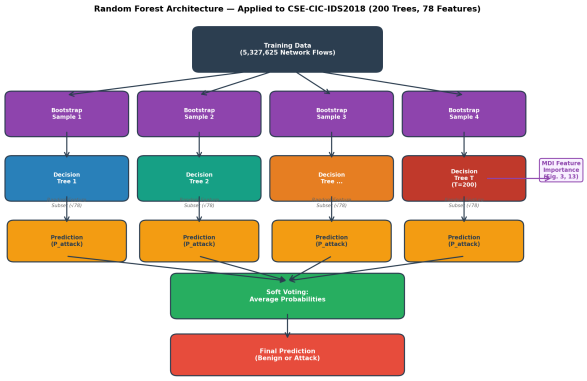

Fig. 3. Random forest architecture applied to CSE-CIC-IDS2018.

Fig. 3 Random Forest architecture applied to CSE-CIC-IDS2018: 200 trees trained on bootstrap samples of 5,327,625 flows, with MDI feature importance extracted and fed into the vulnerability-aware interpretation. Original diagram created for this paper.

C. XGBoost Theoretical Basis

In 2016, Chen and Guestrin [5] proposed a regularized gradient-boosted tree framework, XGBoost. Unlike Random Forest, XGBoost grows trees one after the other, each one "fixing" errors made by the previous one. This sequential architecture is applied in this study (300 trees of maximum depth 6, learning rate 0.1, 80% subsampling) and is shown in Fig. 4. The objective function is a differentiable loss (binary cross-entropy) together with regularization terms that are based on the degree of complexity of the model [5] , [31]. For the current experiment, XGBoost had far fewer false positives compared to Random Forest (760 compared to 14,387). Though regularization of the objective function, limiting the depth of the tree, learning rate, and subsampling may provide some explanations for complexity control [5], the current experiment does not single out the effect of these design principles. Hence, the difference in false positives seen in the current experiment can only be taken as an empirical difference between these models.

As shown in Fig. 4, the XGBoost configuration utilized in CSE-CIC-IDS2018 is demonstrated in Fig. 4 with 300 trees of depth 6, learning rate of 0.1 and 0.8 subsampling for rows and columns. These configuration parameters control model complexity, although the role played by each configuration parameter was not tested experimentally.

D. Soft-Voting Ensemble Theoretical Basis

In soft-voting ensemble, a collection of classifiers' class probability are combined together and the class having maximum average probability is chosen [6]. Ensemble combination of RF and XGBoost probability output is shown in Fig. 5. The systematic review conducted by Tama and Lim [6] showed that the improvement of ensemble is dependent on the complementarity of the models in IDS applications. The ensemble achieves 259 more false positives than standalone XGBoost, supporting Tama and Lim's theoretical claim that when the

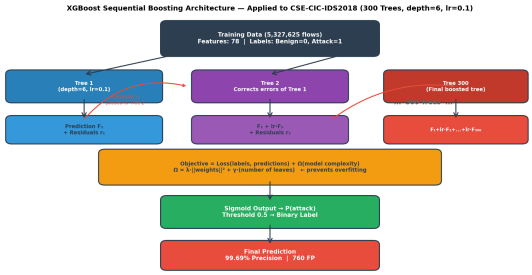

Fig. 4. XGBoost sequential boosting architecture applied to CSE-CIC-IDS2018.

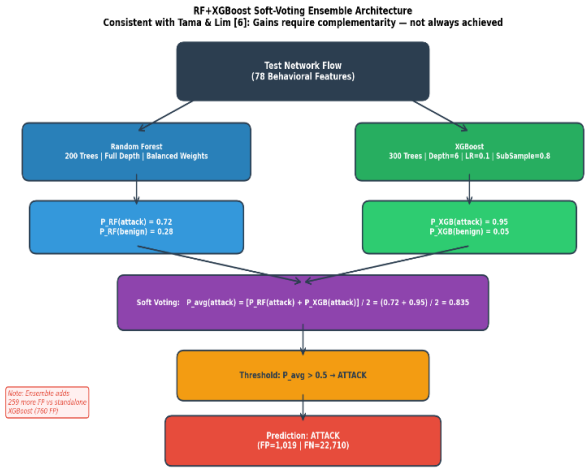

Fig. 5. RF+XGBoost soft-voting ensemble architecture.

two parts are already strong, soft averaging adds noise instead of complementarity.

Fig. 5 RF+XGBoost soft-voting ensemble architecture: averaged probability outputs from both models. The annotation shows the ensemble introduces 259 more false positives than standalone XGBoost, consistent with Tama & Lim [6]. Original diagram created for this paper.

E. Feature Attribution: MDI and SHAP

Two feature-attribution approaches have been analyzed within the scope of the current research, with different functions. The Mean Decrease in Impurity (MDI) score, calculated from the trained RF model, constitutes the primary method for calculating global feature importance scores. The MDI ranking allows one to define the behavioral network-flow characteristics contributing to the RF decision-making the most and formulates the feature set for further vulnerability-focused interpretation.

SHapley Additive exPlanations (SHAP), which offers an addition prediction-level explanation approach based on Shapley value [7], is another method used in this research. In contrast to MDI, which provides an estimation of the feature importance on the whole RF model level, SHAP allows calculating the feature contribution to a particular prediction. However, in the current research implementation, SHAP is

applied only for additional illustrative purposes Fig. 18 rather than serving as the primary attribution technique used in Tables VII and XV. Consequently, conclusions regarding the current research implementation of global feature-importance analysis are based on the RF MDI; the systematic SHAP-based instance-level analysis is the future work [7], [12], [20], [33].

III. Related Work

A. Overview of Existing Work

The study of network intrusion detection has greatly moved from rule-based approaches to data-driven approaches [1], [9], [25], [26]. As shown in Fig. 1, ML-based IDS are the most capable paradigm to deal with new and evolving attack patterns. Prior research is summarized in Table I with an attempt to place the current research in four overlapping literature threads.

B. Critical Analysis

However, tree-based ensemble methods are still very competitive on tabular network-flow data [1], [6], [31]. Ferrag et al. [1] provided a comprehensive overview of deep-learning methods for IDS, noting that the quality of the dataset, the thoroughness of the preprocessing, and the systematic comparison of different methods are key factors to consider before deriving conclusions which all underlie the preprocessing pipeline and identical evaluation conditions applied in the present study. Although deep-learning architecture has been popular recently, recent studies have shown that tree-based models can often give comparable or better performance in problems that involve tabular flow-based data, without the need of a large feature engineering pipeline. Sarhan et al. [14] particularly studied the feature extraction strategies for ML-based IDS for IoT systems and pointed out that it is essential to represent the features in a compact and informative manner, which is why the leakage prone identifiers were removed, and 78 features were kept in the current feature extraction pipeline, since they are informatively meaningful in the IoT environment.

The ensemble gains found by Tama and Lim [6] are dependent on model complementarity which is directly related to the ensemble evaluated in this study as shown in Fig. 5. Ahmed et al. [10] suggested an explainable hybrid ensemble IDS, which has achieved good accuracy in CIC-IDS2017 without relating the model's output to the conditions of underlying IoT vulnerabilities. Adewole et al. [18] put forward an explainable IoT IDS framework by integrating ensemble learning with rule induction and compared Random Forest, AdaBoost, XGBoost, LightGBM, and CatBoost classifiers on CIC-IDS2017 and CICIoT2023 datasets. The performance of the five models for binary classification task on CICIoT2023 dataset demonstrated that XGBoost classifier obtained an accuracy of 98.54%, precision of 98.56%, recall of 98.54%, F1 score of 98.55%, and ROC-AUC of 93.06%. Hence, their results provide relevant external evidence with respect to the IoT field that XGBoost model family still performs well after CSE-CIC-IDS2018. However, they use rule induction for explainability, which is not the case for the present. Balanced class distribution along with feature selection and ensemble ML was explored by Musthafa et al. [16] who directly tackled with the class imbalance problem present in the CSE-CIC-IDS2018 dataset and manifested in the balanced subsample weights utilized in Random Forest in the present work [25]. Ullah et al. [15] used transformer-based transfer learning to cope with the imbalanced IoT traffic and showed that the class distribution issue is a universal problem in IDS architectures.

Theoretically principled feature attribution is provided by the SHAP framework [7] in the explainability thread, which is based on cooperative game theory [33]. Sarker [3] made a case for the importance of XAI in the context of trustworthiness for cybersecurity automation. Explanation-guided feature selection for IoT IDS was presented in [12] by Chen et al., which revealed that per-prediction attribution can boost interpretability and at the same time decrease the model size. To extend explainability to IoT data streams for dynamic online detection scenarios, Bajaber and Alabbadi [17] proposed to add a proportion of the observed data to the features, in addition to the other features. Krishnan et al. [20] investigated the problem of explainable zero-day attack detection, which is particularly relevant to IoT, where new firmware versions and device types continually create new attack surfaces [28].

In the deployment thread, the most implementation-oriented contribution was that of Javed et al. [13] who deployed a tree-based IDS directly on a smart thermostat, and Holubenko et al. [19] explored the concept of decentralized privacy preserving IoT IDS instead of centralized cloud-based analysis [27].

C. Research Gap and Novelty

Review of the aforementioned literature clearly shows that a major deficiency of published IDS works is that models are assessed using aggregate metrics, and there is no explicit correlation of model generated results with the actual IoT vulnerability conditions that would allow the attacks to be detected. This gap is illustrated in Fig. 2, which maps the entire chain of IoT vulnerability to attack behavior to network-flow signature to ML feature a chain which has not been explicitly constructed by existing work (e.g., [10], [12], [18]). In this paper we fill this gap with the introduction of this vulnerability-aware interpretation layer as a first-class contribution.

IV. IoT Vulnerability Characterization

A structured characterization of the class of vulnerabilities to be addressed by the detection framework is necessary before presenting the detection framework. The grounding adds to the binary intrusion detection classifier a vulnerability-based post-processing interpretation mechanism, as explained in Section I-B and illustrated in Fig. 2. The classifier remains an attacker-vs-benign behavior detector, and only the vulnerability requirements are added in the interpretation stage. The IoT security landscape is a unique field with specific systemic vulnerabilities unlike conventional IT infrastructure. Usually, IoT devices do not have sufficient computing power to run a host-based security agent, several devices are deployed behind NAT boundaries, which cannot be directly inspected, and the end user that deploys and manages the devices is usually not a technical user and often does not change default credentials or apply firmware updates [9], [24], [28]. This is because of these structural constraints, network-flow analysis, which shows behavioral patterns at the connection level, is one of

TABLE I. COMPARISON OF REPRESENTATIVE PRIOR WORK AND THE PRESENT STUDY

Study	Dataset	Model	Main Emphasis	Relation to This Work
Ferrag et al. [1]	Multiple IDS	DL review	Survey of DL-based IDS methods, datasets, and challenges	Establishes ML/DL IDS context; motivates tree-based alternatives
Tama & Lim [6]	Cross-benchmark	Ensemble	Ensemble gains depend on model complementarity	Directly motivates ensemble design; prediction confirmed empirically
Saied et al. [9]	IoT IDS literature	Survey/taxonomy	AI-based IoT IDS taxonomy and open research problems	Provides broad IoT IDS context and identifies deployment gaps
Ahmed et al. [10]	CIC-IDS2017	Explainable ensemble	High accuracy combined with hybrid explainable ensemble design	Baseline for external comparison; lacks vulnerability-aware layer
Yaras et al. [11]	CICIoT2023, TON_IoT	Hybrid DL	High-accuracy IDS using deep learning in large-scale IoT settings	Illustrates DL IoT IDS direction; limited deployment optimization
Javed et al. [13]	Smart thermostat	Embedded tree	On-device smart-home IDS using device-readable features	Closest to realistic smart-home deployment; validates tree-based IDS
Adewole et al. [18]	CIC-IDS2017, CICIoT2023	Explainable ensemble	Rule induction linking model decisions to interpretable output	Baseline for comparison; strongly related to transparent IDS
Chen et al. [12]	IoT datasets	DL + XAI	Explanation-guided feature selection reducing model complexity	Supports planned SHAP extension and feature reduction strategy
Musthafa et al. [16]	IoT IDS datasets	Ensemble + FS	Balanced class distribution with feature selection for IoT IDS	Addresses class imbalance; complementary to present preprocessing
Nassif et al. [33]	Multiple	ML anomaly detection	Systematic review of ML for anomaly detection in networks	Supports theoretical grounding of anomaly vs supervised detection
This study	CSE-CIC-IDS2018	RF, XGBoost, soft voting	Comparative evaluation + structured vulnerability-aware interpretation	Integrates performance with feature-to-vulnerability security reasoning

the most practical and scalable detection mechanisms available [13], [14]. Table II characterizes the five most common classes of IoT vulnerabilities, their common attack methods, and the flow signatures they generate. This classification is security and CVE-based vulnerability groups but not directly CVE names; it covers wider IoT security problems whereas particular CVEs like CVE-2014-0160 (Heartbleed) are used as examples. These vulnerable classes are not only defined by the vulnerability but also by the behavior they induce on the generated network traffic, which are directly learnable by the Random Forest and XGBoost models analysed in this study.

These vulnerability classes are not mutually exclusive, so individuals may be classified in both. One attack campaign can take advantage of more than one flaw at once: Mirai botnet [24] used default Telnet credentials (weak credentials), communicated over an exposed port 23 (exposed services) and went on to sustain multiple DDoS attacks due to the number of compromised devices (botnet exploitation). Although the flow-level signatures are different for each class, they are sufficiently distinct to be used as the basis of ML based detection, and several studies have demonstrated that the flow level signatures derived by CICFlowMeter are capable of capturing the signature of attacks at this level [2], [14]. The key is the characterization in Table II, which facilitates the ability of the vulnerability-aware interpretation layer in Section 7 to interpret the results: after the mapping from the vulnerability class to the flow signature is done, the dominant learned features from the RF MDI analysis Fig. 8 may be traced to their most probable security cause.

V. EXPERIMENTAL SETUP AND EVALUATION

A. Objectives and Hypothesis

RQ1: Are tree-based ML models able to accurately distinguish between benign and attack traffic on CSE-CIC-IDS2018 with high accuracy and F1-score (>=0.93) and high ROC-AUC (>0.97)? RQ2: Are there realistic attack manifestations (as shown in Fig. 2) that can be mapped to the above mentioned dominant learned traffic features, and that are relevant to IoT and smart-home environments? The main thesis is that XGBoost's regularization will produce significantly fewer false positives than will Random Forest.

B. Literature Selection Criteria

The literature selection was done systematically. The formal inclusion and exclusion criteria used are shown in Table III. IEEE Xplore, Scopus, ACM Digital Library, ScienceDirect and SpringerLink are among the databases searched. Finally, the paper set of 35 papers was chosen according to the criteria presented in Table III.

C. Dataset Description

The Canadian Institute for Cybersecurity (CIC) created the CSE-CIC-IDS2018 [2] benchmark, which is a large-scale cyber-defense benchmark developed with the Communications Security Establishment (CSE). It records seven different attack scenarios that are played out across multiple days in a controlled network environment, generating network traffic across brute force, botnet, DoS, DDoS, Web attacks, infiltration, and Heartbleed-related scenarios. This data set was collected using CICFlowMeter, a bidirectional network flow tool which collects 80 statistical features for each connection (inter-arrival

TABLE II. IOT VULNERABILITY CLASSES, ATTACK VECTORS, AND NETWORK-FLOW SIGNATURES

Vulnerability Class	Common Examples	Attack Vector	Typical Network-Flow Signature
Weak / default credentials	Factory passwords, shared credentials, no lockout policy	Brute-force login, credential stuffing, dictionary attacks	High-frequency short-lived flows, repeated SYN packets, abnormal packet counts
Exposed services / open ports	Telnet on port 23, unpatched SSH, open HTTP admin panels	Botnet C2, infiltration, port scanning	Persistent low-frequency flows, odd inter-arrival timing, directional asymmetry
Outdated / unpatched firmware	Heartbleed (CVE-2014-0160), known CVEs in router firmware	DoS, DDoS, protocol exploitation	High packet rates, large byte volumes, sustained traffic bursts
Weak web-layer protection	No input validation, missing CSRF tokens, outdated frameworks	SQL injection, XSS, malicious HTTP requests	Irregular request-response timing, direction-specific anomalies
Insecure communication protocols	Unencrypted MQTT, plain HTTP, deprecated TLS	Man-in-the-middle, traffic interception, replay attacks	Clear-text payload patterns, unusual protocol-port combinations

TABLE III. INCLUSION AND EXCLUSION CRITERIA FOR LITERATURE SELECTION

Criterion	Inclusion	Exclusion
Publication type	Peer-reviewed journal articles and conference papers	Blog posts, grey literature, non-peer-reviewed reports
Publication year	2019–2026; seminal pre-2019 works (RF [4], XGBoost [5]) included	Papers before 2019 unless seminal or foundational
Domain relevance	IDS, IoT security, ML-based traffic classification, XAI in security	Papers unrelated to network security or traffic classification
Language	English-language publications only	Non-English publications without English translations
Methodology	Papers with validated experimental results on named benchmark datasets	Papers without empirical validation or reproducible results
Dataset availability	Papers using publicly available or well-documented benchmark datasets	Papers using private or undisclosed datasets without metadata
Venue quality	IEEE, ACM, Springer, Elsevier, MDPI, or equivalent Q1/Q2 peer-reviewed venues	Low-impact, predatory journals, or papers without documented peer review

TABLE IV. DATASET SUMMARY BASED ON THE IMPLEMENTED PIPELINE

Item	Value / Description
Dataset	CSE-CIC-IDS2018
Source	Canadian Institute for Cybersecurity (CIC) and Communications Security Establishment (CSE)
Task type	Binary classification benign (0) vs. attack (1)
Feature extraction tool	CICFlowMeter generates 80 bidirectional flow features per connection
Original flow features	80 CICFlowMeter-derived flow features
Final encoded feature space	78 features (after leakage removal and categorical encoding)
Training set size	5,327,625 flows (80% stratified split)
Test set size	1,331,907 flows (20% stratified split)
Attack families	Botnet, brute force, DoS, DDoS, infiltration, web attacks, Heartbleed-related scenarios
Class distribution	Imbalanced benign traffic dominates; handled via balanced class weighting in RF
File format	Apache Parquet (columnar format for efficient large-scale processing)
Split strategy	Stratified 80/20 train-test split class proportions preserved across both partitions

times, packet sizes, window sizes, number of flags, number of bytes, etc.) without generating any raw packet payloads [2]. The flow-level is particularly useful for IoT IDS, as it is less computing intensive than deep packet inspection and still provides behaviour information to distinguish between benign and malicious communication patterns [13] and [14].

One key point to keep in mind here is that CSE-CIC-IDS2018 is not a specific smart-home or IoT device capture. The traffic was generated in a controlled enterprise-like network environment and not in the context of a residential IoT deployment. But, these attack families are exactly same with the behaviors targeted on IoT devices in smart-home environments as characterized in Table II and literature of vulnerabilities [9], [24], [28]. Consequently, the word "smart-home-relevant" is used in the manuscript in order not to assume that CSE-CIC-IDS2018 is a smart-home IoT deployment. The dataset is extensively employed as a benchmark in the IDS literature [1], [6], [10], [18] and is hence suitable for external comparison. Table IV summarizes the entire set of data.

D. Preprocessing and Cleaning

The preprocessing pipeline is based on a supervised ML workflow specifically designed for intrusion data in network flows. The system workflow is shown in Fig. 6, in which data is processed from raw data to vulnerability-aware interpretation. Each of the preprocessing steps and their reasons is summarized in Table V.

E. Leakage-Aware Feature Removal

In order to avoid data leaks and endpoint memorization, the fields that resemble identifiers were eliminated prior to training. The removed features included flow identifiers, source and destination IP addresses, source and destination port identifiers, and other timestamp-related features found in the original flow. These features are potentially prone to leakage as they provide the information about the identity of endpoints or collection sessions and not the true behavior of the network flow. If certain source and destination addresses are frequently seen in attack cases, the classifier will receive high results in test phase due to memorization of the label-endpoint relations.

The criterion that guided the selection of features in this paper was thus functional and not merely statistical in nature, whereby a feature was removed if its main purpose was to detect an endpoint, connection, or period of acquisition rather than the behavior of the flow itself. On the other hand, characteristics of behavior, such as Flow Duration, number of packets, length of packets, statistics on inter-arrival times, window size, and flow rate, were included since they depict the process of communication and may be useful for use in a novel situation.

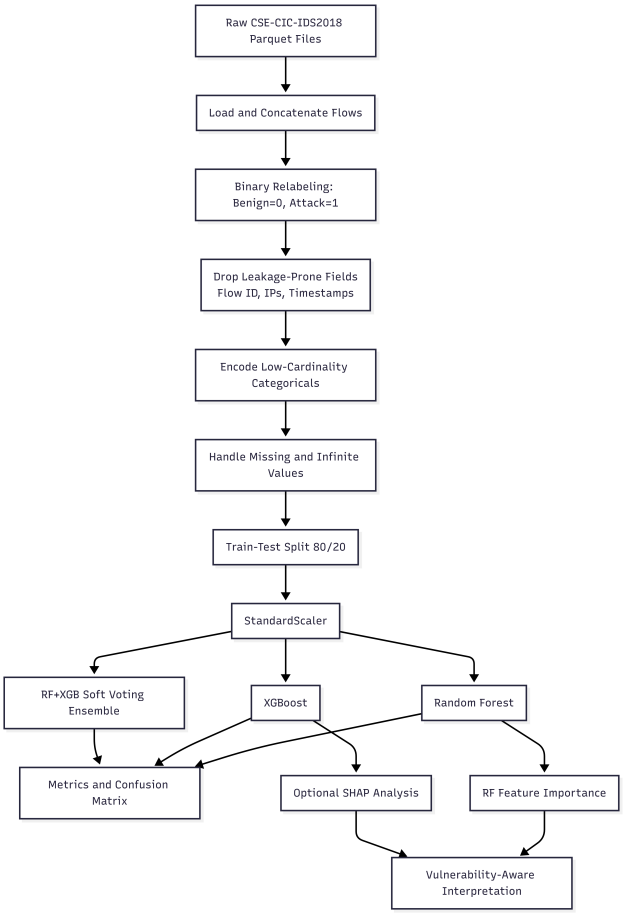

Fig. 6. Overall workflow of the proposed vulnerability-aware intrusion detection framework.

Fields which suffer from leakage issues have been removed from the predictor matrix prior to modeling, while the class label is kept for prediction purpose alone. This helps in minimizing memorization of endpoint and identifier information, and also promotes a behavior-based network flow evaluation. But since the evaluation strategy used here is stratified random 80/20, it does not ensure that the attacks sessions, flows, time periods, and campaigns between the training and test datasets are independent of each other. This implies that the test results are indicative of performance on data that has been seen but held out, and not on unseen sessions and campaigns.

As shown in Fig. 6 Overall workflow of the proposed vulnerability-aware intrusion detection framework. Taken directly from the implemented framework.

F. Model Training and Configuration

Three models are considered. The RF architecture is shown in Fig. 3, and XGBoost architecture in Fig. 4 and ensemble in Fig. 5. The complete hyperparameter information can be found in Table VI.

G. Evaluation Metrics and Baselines

Accuracy, precision, recall, F1-score, ROC-AUC and confusion matrices are used to measure model performance.

TABLE V. PREPROCESSING STEPS AND IMPLEMENTATION DETAILS

Step	Purpose and Implementation Detail
Parquet aggregation	All Parquet segment files loaded and concatenated into a single unified DataFrame for consistent downstream processing
Binary relabeling	Multiclass labels collapsed to binary: benign $\rightarrow$ 0, all attack categories $\rightarrow$ 1. Enables a focused binary detection task.
Leakage-aware identifier removal	Identifier and acquisition related features such as flow IDs, source and destination IPs, source and destination ports, and time stamp related features were not included in the analysis to prevent memorization of hosts or sessions associated with attacks. Only behavior based flow features were used to predict attacks.
Categorical encoding	Low-cardinality categorical fields (e.g., protocol type) label-encoded into machine-readable integer form
Missing / infinite handling	np.inf and -np.inf replaced with NaN; all NaN entries filled with zero to eliminate numerical instability
Feature scaling	All features standardized via StandardScaler (zero mean, unit variance) for pipeline consistency
Stratified split	Stratified 80/20 train-test split ensures identical class-proportion representation across both partitions

TABLE VI. MODEL HYPERPARAMETER CONFIGURATIONS

Parameter	Random Forest [4]	XGBoost [5]
Number of estimators	200 trees	300 trees
Maximum tree depth	Unlimited (None)	6 (regularization to control over-fitting)
Learning rate	N/A (bagging-based)	0.1 (shrinkage per tree contribution)
Row subsampling	Bootstrap sampling ($\sim$63.2%)	0.8 - 80% of training rows per tree
Column subsampling	sqrt(n_features) per split	0.8 - 80% of features per tree
Objective function	Gini impurity / entropy	binary:logistic (outputs calibrated class probabilities)
Evaluation metric	Out-of-bag (OOB) error	Log-loss (logloss)
Class imbalance handling	class_weight= 'balanced_subsample'	Implicit via scale_pos_weight in loss
Ensemble combination	Majority voting / probability averaging	Soft-voting: averaged probabilities from RF + XGBoost
Implementation library	scikit-learn RandomForestClassifier	xgboost XGBClassifier

Precision directly controls analyst alert fatigue, recall controls the coverage of the security, F1-score controls a balance of both, and ROC-AUC controls the discriminative ability of the security without a threshold. Operationally critical measures of false positive and false negative counts are reported explicitly. Training times and memory consumption are not instrumented and are known to be a limitation that is addressed in Table XVII. Ahmed et al. [10] and Adewole et al. [18] are the sources of the baseline comparisons.

H. Structured Vulnerability-Oriented Interpretation

The structured interpretation layer maps the prominent features of the behavioral network flow to potential attacks and vulnerabilities conditions, as described in Table VII and presented in Fig. 2. Explainable intrusion detection research stresses the importance of the linkage between the behavior of the model and the interpretable security evidence [3], [10], [18]. The attribution methodologies based on SHAP

TABLE VII. STRUCTURED VULNERABILITY-TO-ATTACK MAPPING FRAMEWORK

Vulnerability Class	Likely Attack Manifestation	Representative Flow-Level Behavior	Relevant Dominant Features
Weak / default credentials	Brute-force, repeated login probing	Repeated short-lived flows, retries, abnormal packet-count patterns	Total Fwd Packets, Fwd Packets Length Total, Fwd Packet Length Mean
Exposed services / open ports	Botnet C2, infiltration, command-and-control	Odd timing regularity, unusual duration, directional traffic imbalance	Flow Duration, Fwd IAT Std, Flow IAT Max, Init Fwd Win Bytes
Outdated / unpatched systems	DoS / DDoS exploitation and Heartbleed	High packet rates, large byte volumes, sustained traffic bursts	Fwd Packets/s, Subflow Fwd Bytes, Fwd Packet Length Max
Weak web-facing protection	Web attacks, SQLi, XSS	Irregular request-response timing, direction-specific anomalies	Flow IAT Mean, Flow IAT Min, Init Bwd Win Bytes

TABLE VIII. RANDOM FOREST PERFORMANCE SUMMARY

Metric	Value
Accuracy	0.9725
Precision	0.9433
Recall (highest among models)	0.9165
F1-Score	0.9302
ROC-AUC	0.9797
False Positives	14,387
False Negatives	22,223

and similar techniques offer a principled way of attributing influential features at the model or prediction level [7], [12], [17]. In this paper, the attributed behavioral features are further interpreted in terms of the potential vulnerability conditions of the IoT devices, in line with the explainable-IoT security reasoning previously discussed [18], [33]. Thus, the presented associations should be considered as semantic interpretations from the literature, but not as vulnerability labels learned by the classifier.

VI. RESULTS AND ANALYSIS

A. Random Forest

As shown in Fig. 7 and Table III, Random Forest [4] achieved accuracy 0.9725, precision 0.9433, recall 0.9165, F1-score 0.9302, and ROC-AUC 0.9797, with confusion matrix 1,051,415 TN / 14,387 FP / 22,223 FN / 243,882 TP. These results validate RF as a solid foundation for detection with the highest recall value, directly addressing RQ1.

As shown in Fig. 7 Confusion matrix of the Random Forest model on the CSE-CIC-IDS2018 test set (Table VIII). Produced by the actual model evaluation.

The 14,387 false positives represent a substantial operational burden. As shown in the figures: In the case of classifiers 3 and 13, the feature-importance analysis shows that timing related and flow-volume features are the most discriminative ones. Among the top-ranked features, Init Fwd Win Bytes, Flow Duration, Init Bwd Win Bytes, Fwd IAT Std and Flow IAT Max are behavioural features and the results indicate that the model is reacting to meaningful traffic dynamics. A mapping of these features and IoT vulnerabilities are presented in Fig. 2.

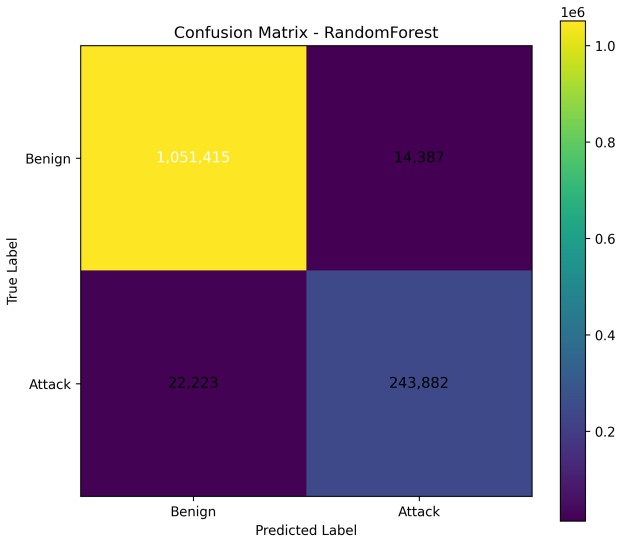

Fig. 7. Confusion matrix of the Random Forest model on the CSE-CIC-IDS2018 test set.

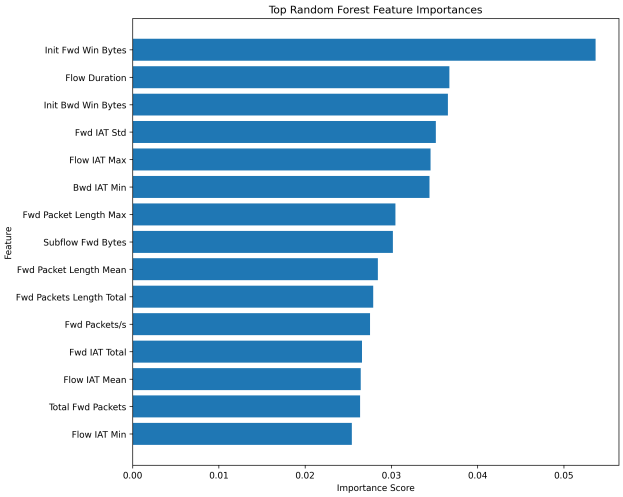

Fig. 8. Confusion matrix of the random forest model on the CSE-CIC-IDS2018 test set.

As shown in Fig. 8 Top 15 most important features (RF MDI scores). Produced by the actual model evaluation.

B. XGBoost

As shown in Fig. 9 and Table IX, XGBoost [5] achieved the best overall performance: accuracy 0.9822, precision 0.9969, recall 0.9139, F1-score 0.9536, and ROC-AUC 0.9907 with confusion matrix 1,065,042 TN / 760 FP / 22,900 FN / 243,205 TP. However, the number of false positives 760 is almost 19 times less compared to the number of the evaluated RF setup. This provides empirical performance gap between the two setups, but it does not indicate the source of the problem. The gap may be due to differences in regularization, maximal depth of trees, subsampling, handling of class imbalance, and other hyperparameters as shown in Fig. 4.

As shown in Fig. 9 Confusion matrix of the XGBoost

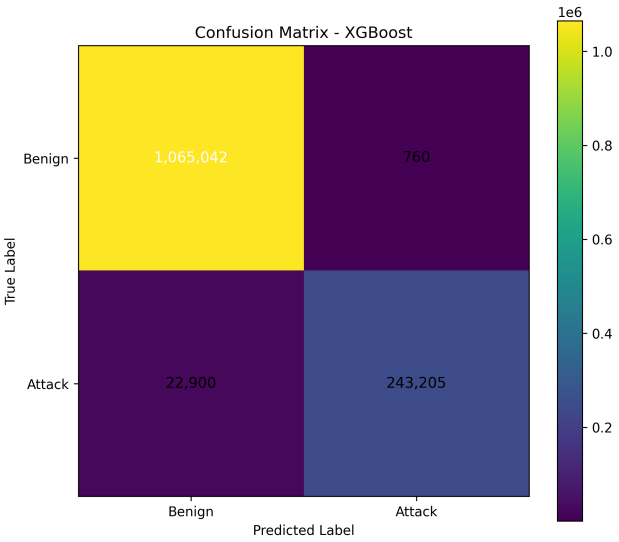

Fig. 9. Confusion matrix of the XGBoost model on the CSE-CIC-IDS2018 test set.

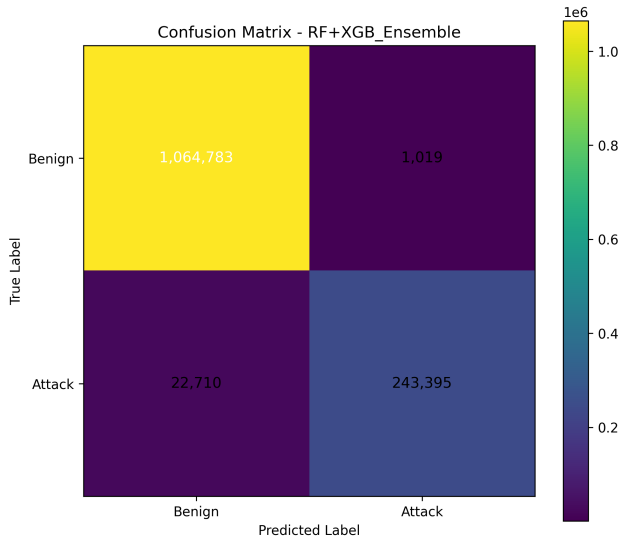

Fig. 10. Confusion matrix of the RF+XGBoost soft-voting ensemble.

TABLE IX. XGBOOST PERFORMANCE SUMMARY

Metric	Value
Accuracy	0.9822
Precision (highest among models)	0.9969
Recall	0.9139
F1-Score (highest among models)	0.9536
ROC-AUC (highest among models)	0.9907
False Positives (lowest among models)	760
False Negatives	22,900

TABLE X. RF+XGBOOST ENSEMBLE PERFORMANCE SUMMARY

Metric	Value
Accuracy	0.9822
Precision	0.9958
Recall	0.9147
F1-Score	0.9535
ROC-AUC	0.9896
False Positives	1,019
False Negatives (lowest among models)	22,710

model on the CSE-CIC-IDS2018 test set. Produced by the actual model evaluation.

C. RF+XGBoost Soft-Voting Ensemble

As shown in Fig. 10 and Table X, the ensemble achieved accuracy 0.9822, precision 0.9958, recall 0.9147, F1-score 0.9535, and ROC-AUC 0.9896, with confusion matrix 1,064,783 TN / 1,019 FP / 22,710 FN / 243,395 TP. The ensemble adds 259 more false positives (as listed in Fig. 5) than standalone XGBoost, which is empirically the same as Tama and Lim [6].

As shown in Fig. 10 Confusion matrix of the RF+XGBoost soft-voting ensemble. Produced by the actual model evaluation.

D. Comparative Summary

A comprehensive side-by-side comparison is offered in Table XI for all eight metrics reported. The best result for each metric is highlighted in bold and marked with an asterisk (*). These results show an obvious hierarchy in terms of operational IDS performance. Based on metrics most important to alert-driven deployment, XGBoost performs best with the highest precision (0.9969), highest F1-score (0.9536), highest ROC-AUC (0.9907) and lowest number of false-positive alerts (760). Random Forest has the highest recall (0.9165), but it is only 0.0026 higher than XGBoost, which is virtually indifferent given the 14,387 false positive count of Random Forest and 760 false positive count of XGBoost. The ensemble has the fewest false negatives (22,710) but at the price of 259 more false positives versus XGBoost and a marginal recall gain of 0.0008 over XGBoost, which is not adequate to offset the loss of precision. The results are internally consistent. In contrast, XGBoost's depth limited trees (maximum depth is 6) and explicit regularization, prevent the spurious overfitting of individual decision boundaries, which leads to RF generating false positives at high tree depth without depth constraint. The ensemble's results were not improved from XGBoost which is consistent with the effect of soft-voting that was shown in Fig. 5: averaging a signal with higher precision (XGBoost) with a signal with lower precision (RF) inevitably reduces the precision regardless of the marginal improvements in the recall. These observations are supported by a set of graphical metric comparisons, which are directly generated by the implemented framework, shown in Fig. 11, Fig. 12, Fig. 13, Fig. 14, Fig. 15, Fig. 16, considering the five evaluation dimensions. (* = best value per metric; Acc=Accuracy, Prec=Precision, FP=False Positives, FN=False Negatives)

E. Comparison with Prior Studies

The results presented here are compared to two similar published studies in Table XII. Ahmed et al. [10] reported best accuracy 98.79% on CIC-IDS2017. However, in the present study, the accuracy is obtained 98.22% on the more

TABLE XI. OVERALL COMPARISON OF MODEL PERFORMANCE

Model	Acc.	Prec.	Recall	F1	ROC-AUC	FP	FN
Random Forest [4]	0.9725	0.9433	**0.9165***	0.9302	0.9797	14,387	22,223
XGBoost [5]	**0.9822***	**0.9969***	0.9139	**0.9536***	**0.9907***	**760***	22,900
RF+XGBoost Ensemble	0.9822*	0.9958	0.9147	0.9535	0.9896	1,019	**22,710***

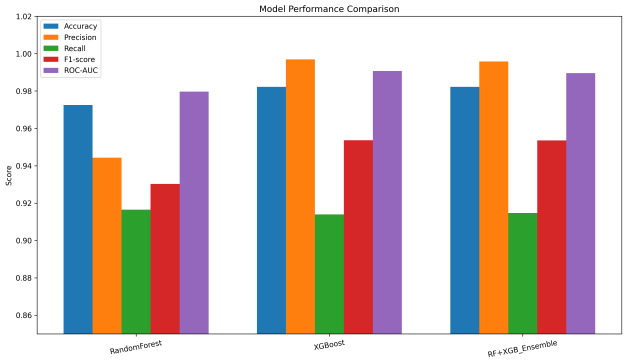

Fig. 11. Overall model performance comparison. Produced by the implemented framework.

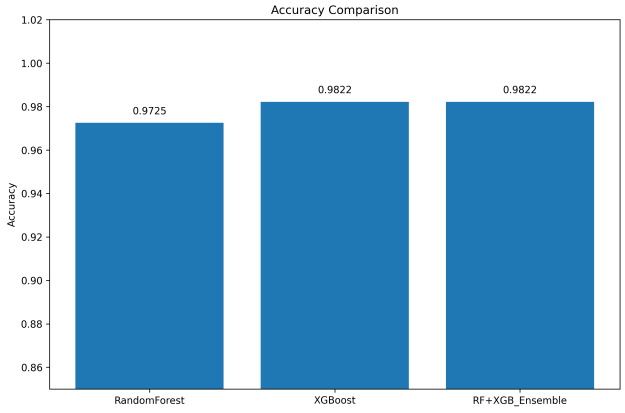

Fig. 12. Accuracy comparison. Produced by the implemented framework.

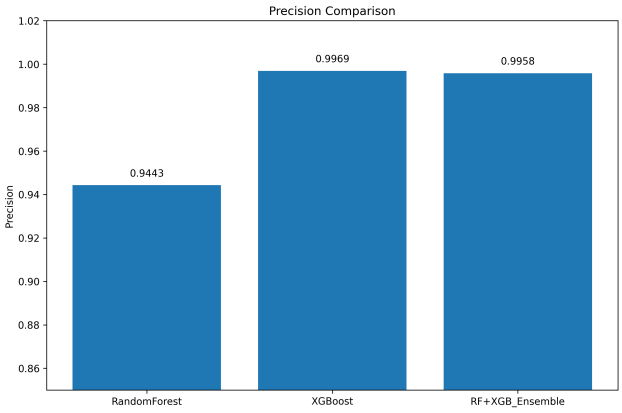

Fig. 13. Precision comparison. Produced by the implemented framework.

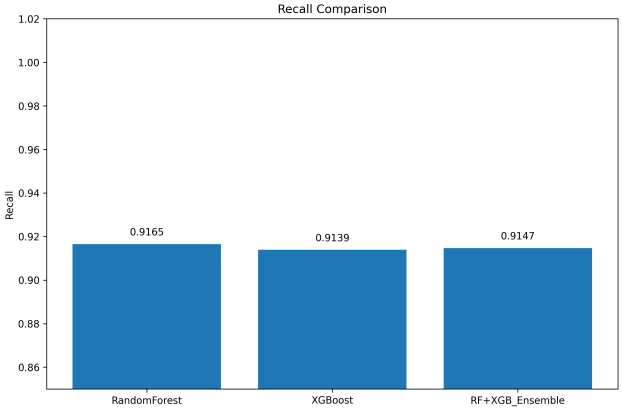

Fig. 14. Recall comparison. Produced by the implemented framework.

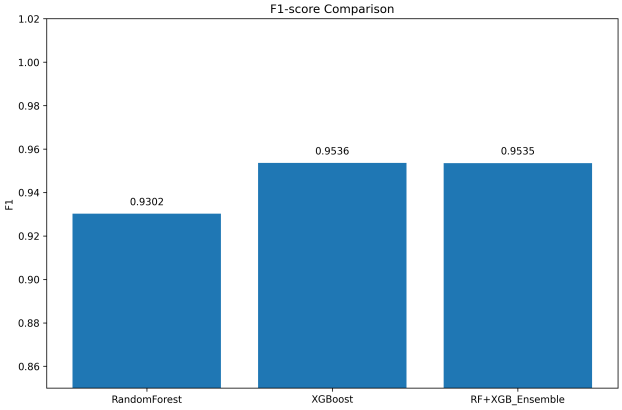

Fig. 15. F1-score comparison. Produced by the implemented framework.

challenging CSE-CIC-IDS2018 benchmark and the precision is achieved 99.69% which is much higher. Adewole et al. [18] tested XGBoost on CIC-IDS2017 (99.91%) and CICIoT2023 (98.54%, 93.06% ROC-AUC). In the present study, the ROC-AUC is 99.07% which is significantly better than the IoT-oriented result. One major difference lies in the capacity for vulnerability-aware interpretation chain shown in Fig. 2, which does not exist in either of the reference studies.

F. External IoT Benchmark Comparison

In order to present further IoT-relevant evaluation context, apart from the benchmark used in this work, the performance of the proposed XGBoost classifier is evaluated against the independently published XGBoost performance of Adewole et al. [18] on the IoT-relevant CICIoT2023 dataset. Such a comparison makes sense especially considering that both works use XGBoost for binary intrusion detection, but on different datasets. The results of such a comparison are presented in Table XIII. This comparison should be considered external benchmarking data rather than a cross-dataset experiment itself, since the datasets differ significantly.

Table XIII illustrates how the findings of the current research are contextualized within the body of recent literature related to explainability and ensemble-based IDS research. The

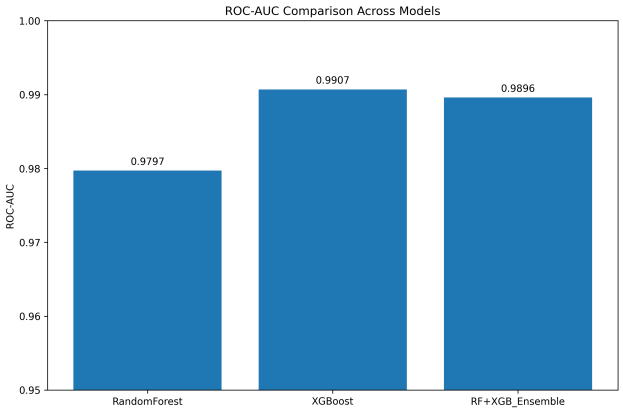

Fig. 16. ROC-AUC comparison. Produced by the implemented framework.

TABLE XII. COMPARATIVE EVALUATION WITH SIMILAR STUDIES

Study	Dataset	Main Models	Best Reported Performance	Comparison with This Study
Ahmed et al. [10]	CIC-IDS2017	Hybrid adaptive ensemble IDS	Best accuracy ∼98.79%	Present: 98.22% accuracy, 99.69% precision superior reliability on harder benchmark
Adewole et al. [18]	CIC-IDS2017 & CICIoT2023	RF, AdaBoost, XGBoost, LightGBM, CatBoost	XGBoost: 99.91% on CIC-IDS2017; 98.54% acc, 93.06% ROC-AUC on CICIoT2023	Present: 98.22% acc, 99.07% ROC-AUC stronger ROC-AUC than IoT-oriented result
Present study	CSE-CIC-IDS2018	RF, XGBoost, RF+XGBoost soft voting	XGBoost: 98.22% acc, 99.69% prec, 91.39% recall, 95.36% F1, 99.07% ROC-AUC	Competitive with recent IDS literature; adds structured vulnerability-aware interpretation

work by Adewole et al. [18] constitutes especially relevant external evidence, since their framework was validated in an IoT-oriented CICIoT2023 dataset. Employing the XGBoost algorithm to implement a binary classification model, they attained 98.54% accuracy, 98.56% precision, 98.54% recall, 98.55% F1-score, and 93.06% ROC-AUC.

The current research achieves 98.22% accuracy, 99.69% precision, 91.39% recall, 95.36% F1-score, and 99.07% ROC-AUC through application of XGBoost on CSE-CIC-IDS2018 dataset. While the results cannot be directly compared because of the differences in the datasets used in two studies and in their respective experimental frameworks, the independent CICIoT2023 benchmark findings constitute additional evidence that XGBoost remains an effective algorithm in the IoT-oriented traffic environment. This is relevant to the current research since it reduces dependency on performance evidence based on one benchmark.

However, the difference between the two studies is mainly on the interpretation layer, where Adewole et al. [18] use rule induction to create interpretable reasons for model decision-making, while the current approach makes the connection between important behavioral network flow characteristics and possible attack behaviors and vulnerable IoT. Therefore, the external CICIoT2023 data validates the appropriateness of using the chosen models, while the current contribution is more concerned with extending the interpretation towards vulnerability-aware security reasoning.

G. External Evidence for Cross-Dataset Applicability

Since the main experiments in this study utilize CSE-CIC-IDS2018, independent evidence from the benchmarks specific to the IoT domain should be taken into account regarding the generalizability of the chosen detection method. In particular, the research by Adewole et al. [18] evaluates the performance of XGBoost on CICIoT2023 and obtains 98.54% accuracy, 98.56% precision, 98.54% recall, 98.55% F1-score, and 93.06% ROC-AUC for binary intrusion detection. Thus, it becomes clear that even in a fundamentally different benchmark, the model family is able to retain strong classification performance.

While the obtained evidence cannot be considered direct cross-dataset verification of the implementation created in the current study due to differences between the datasets, preprocessing techniques, feature sets, and experiment setups, it still adds additional evidence to the claim about good performance of the same model family in the IoT traffic environment, which makes it a good choice for the detection part of the suggested solution.

VII. EXPLAINABILITY AND VULNERABILITY-AWARE DISCUSSION

In this section, learned patterns are mapped to plausible vulnerability scenarios for security and CVE to show how machine learning based traffic analysis can be linked to device security context. The entire chain from IoT vulnerability to detection feature is shown in Fig. 2. This mapping is not causal: models are not designed to detect vulnerabilities directly, but identify traffic patterns that are able to infer vulnerabilities-enabled behaviors [3], [7], [12], [17], [20], [33]. This distinction is important, as the models return a binary label (benign or attack) by learning the statistical patterns among the 78 behavioral features, which they receive, but without having any information on the underlying vulnerability class. The vulnerability-aware interpretation is a post hoc explanatory model based on the understanding of the behavior of the different types of attacks at the network flow level. This directly answers RQ2.

The top-ranked features from the Random Forest MDI analysis (Fig. 8) Init Fwd Win Bytes, Flow Duration, Init Bwd Win Bytes, Fwd IAT Std, Flow IAT Max, Fwd Packet Length Max, Subflow Fwd Bytes, Fwd Packets/s, Bwd IAT Min, Fwd Packet Length Mean, Fwd Packets Length Total, Fwd IAT Total, Flow IAT Mean, Total Fwd Packets, and Flow IAT Min are all behavioral attributes and not explicit identifiers.

As shown in Fig. 17 presents the conceptual mapping between vulnerability classes, their attack manifestations, and the representative network-flow indicators.

TABLE XIII. EXTERNAL IOT BENCHMARK COMPARISON OF XGBOOST PERFORMANCE

Study / Dataset	Model	Accuracy	Precision	Recall	F1-score	ROC-AUC
Adewole et al. [18] / CICIoT2023	XGBoost	98.54%	98.56%	98.54%	98.55%	93.06%
Present Study / CSE-CIC-IDS2018	XGBoost	98.22%	99.69%	91.39%	95.36%	99.07%

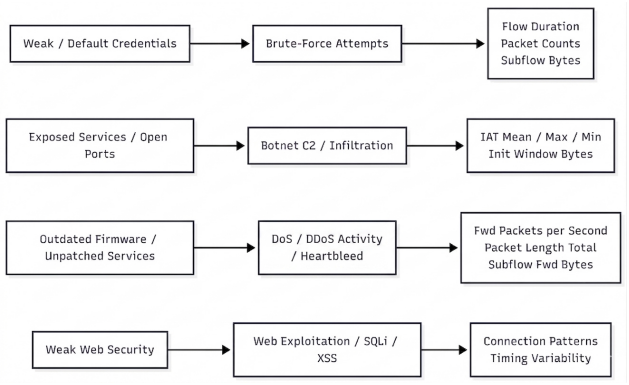

Fig. 17. Conceptual mapping between smart-home IoT vulnerabilities, attack behaviors, and network-flow indicators.

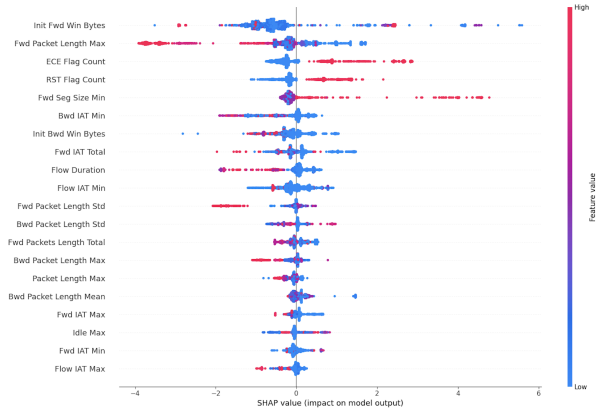

Fig. 18. SHAP feature attribution analysis produced.

As shown in Fig. 17 Conceptual mapping between smart-home IoT vulnerabilities, attack behaviors, and network-flow indicators. Produced by the implemented framework.

Weak or default credentials [9], [24] appear as repeated short-lived sessions, login retries and abnormal packet-count patterns such as Total Fwd Packets, Fwd Packet Length Mean, and Fwd Packets Length Total. In a brute force attack scenario, an attacker tries different combinations of credentials, creating a short-lived TCP connection each time that fails to successfully log in. All of these attempts in a short period of time create a unique high rate/low duration flow signature [29]. Persistent flows that are generated by exposed services and open ports have a directionality that is asymmetric, which agrees with Flow Duration, Fwd IAT Std, Flow IAT Max, and Init Fwd Win Bytes [13]. One of the most notable features of botnet C2 communication is its pseudo-periodic inter-arrival structure: bots periodically contact the C2 server, resulting in a rhythmic pattern in the Flow IAT statistics, which is not easy to obtain by coincidental benign activity. Volume (high packet rates mapping onto Fwd Packets/s, Subflow Fwd Bytes and forward packet length max values) is a characteristic of DoS and DDoS traffic, while outdated or unpatched systems generate high packet rates. Weak web-facing protection occurs when the timing anomalies are irregular, which is identified by the Flow IAT Mean, Flow IAT Min, and Init Bwd Win Bytes [9], [12]. Examples of SQL injection and XSS attacks are HTTP requests with non-regular payload structures and response patterns that deviate from the normal statistical distribution of requests/responses at the flow level.

As can be seen in Fig. 18, the implemented framework generates the SHAP analysis. SHAP values [7] are shown as an explanation per prediction in addition to the global MDI ranking in Fig. 8. MDI provides insight into what features are most influential at a summary level of all prediction, while SHAP provides insight to what features are most influential at the individual level of prediction. A flow that is flagged by the model as malicious can be caused by an unusually high Fwd Packets/s (which is typical of a DoS pattern), or an abnormally long Flow Duration with periodic Fwd IAT Std (which is typical of C2 timing), and SHAP quantifies these contributions at the instance level. SHAP is also used in addition to the global RF-MDI ranking of features based on its predictive results. Here, the direction and magnitude of the SHAP values show the level of contribution of the corresponding network flow features to the result of the model prediction. The attributions are later interpreted using the vulnerability-aware mapping layer, where major behavioral features are linked to possible attack patterns and IoT vulnerability classes. Most importantly, the mapping is interpretative and not causal, as the model identifies the traffic behavior but cannot pinpoint a CVE or an existing vulnerability on the endpoint. Table XIV lists the detailed interpretation of each feature-group.

As shown in Fig. 18 SHAP feature attribution analysis produced by the implemented framework, providing instance-level explainability complementary to the global RF MDI rankings.

VIII. DISCUSSION

A. Interpretation of Results

From the experimental results, three important observations can be made. The first is a basic difference in operation: the number of false-positives is 760 for XGBoost and 14,387 for RF. While the accuracy of the two models is close to each other (0.9822 and 0.9725), there is a subtle difference in operationally key numbers: RF produces 18.9× more false alarms per 1.3 million flows. In a real-life IDS deployment, which processes high volume IoT network traffic, this means thousands of extra analyst investigations per day, which takes time and in the real world can result in real attack alerts being deprioritized or missed [9], [13]. The reason is structural: RF is a tree based algorithm with an unconstrained depth that can overfit local decision boundaries that are specific to the flow characteristics at the training time, and thus result in false

TABLE XIV. FEATURE-GROUP INTERPRETATION AND ATTACK-BEHAVIOR LINKAGE

Feature or Feature Group	Operational Meaning	Possible Attack-Behavior Interpretation
Init Fwd / Bwd Win Bytes	TCP window size at session start	Abnormal window sizes may indicate connection-setup anomalies, aggressive scanning, or botnet session initiation
Flow Duration; Flow IAT Mean/Min/Max	Temporal persistence and inter-arrival timing	Periodic C2 timing, brute-force retry loops, and burst-based DoS produce distinctive temporal signatures
Fwd Packet Length Mean/Max; Fwd Packets Length Total	Forward traffic volume and packet sizes	Flooding, abnormal HTTP request sizes, and DoS bursts manifest as unusual packet-length distributions
Fwd Packets/s; Total Fwd Packets	Packet transmission rate and count	DoS attacks and automated probing tools generate distinctively high packet rates
Subflow Fwd Bytes; Fwd IAT Total	Directional byte volume over subflow segments	Coordinated botnet activity and multi-stage attack sessions produce anomalous subflow-level patterns
Bwd IAT Min; Bwd IAT Std	Backward inter-arrival timing variability	Server response irregularities may indicate C2 response patterns or filtered/throttled connections

positive on benign test flows. The regularization mechanism of XGBoost (shown in Fig. 4) discourages this complexity and generates a model with high recall and a significant control of the generation of false alarms, as theoretically shown in [5].

Second, the soft-voting ensemble result (Fig. 5) verifies the theoretical prediction of Tama and Lim [6] that the ensemble gains need model complementarity. The difference between the precision of RF (0.9433) and XGBoost (0.9969) is 5.36 percentage points, which is a significant amount and indicates that RF and XGBoost are not making similar types of errors. RF has many more false positives and its probability estimates are averaged with a more conservative ones from XGBoost, resulting in a combined output with the false positive tendency of RF, generating 1,019 FP as opposed to XGBoost's 760 FP. In an operationally realistic IDS scenario, this false-positive penalty is not justified by the ensemble's marginal recall improvement (+0.0008). Thirdly, the vulnerability-aware interpretation (Fig. 2) provides an extra level of value that cannot be achieved with the use of the aggregate metrics. A strong reaction to Fwd Packets/s and Subflow Fwd Bytes means it is probably a model that reacts to flooding behavior; a strong reaction to Flow Duration and Fwd IAT Std means that it is likely a model that reacts to C2 timing; a strong reaction to Total Fwd Packets and Fwd Packet Length Mean means that it is likely a model that reacts to brute-force probing behavior [3], [10], [12] These interpretations help facilitate analyst response – not undirected investigation.

The excellent performance of XGBoost that was noted in the current set of experiments is also supported by the IoT-focused evidence from outside sources. According to Adewole et al., [18], XGBoost reached 98.54% accuracy and 98.55% F1-score when applied to CICIoT2023, which means that this machine learning method performed best among all the tested ensemble classifiers. The agreement between the independently carried out experiments that focused on the same intrusion detection tasks can serve as an additional reason to choose XGBoost as a classifier in the framework.

B. Practical and Theoretical Implications

From a practical point of view, XGBoost's accuracy of 99.69% translates to 99.7% of alerts being true attacks, thus directly decreasing the burden of investigation for analysts. In an IoT context, where security personnel are few and budgets are constrained [9], [13], [28], this is more than a numerical difference it represents the gap between a manageable and an unmanageable alert queue. Moreover, the interpretation layer (Fig. 2) based on a vulnerability analysis renders the system actionable in an operational context, and this is not possible by pure metric reporting. When a security operator receives an alert for a short-lived, high-packet-count flow, the operator can investigate whether the associated device uses default factory credentials [32], [3]. An alert indicating the presence of periodic flows, which have stable inter-arrival times, can guide the analyst to investigate for any unauthorized periodic connections that may be associated with botnet C2 activity. This conversion of the classification output to security reasoning is the main practical contribution of the vulnerability-aware layer.

Theoretically, the fact that dominant learned features are behavioral – as is always revealed in Fig. 8 verifies the basic rule of flow-based ML-IDS: models are trained based on the behavior of features, which means they learn traffic dynamics, not endpoint specific artefacts [4], [5]. Many IDSs have identifier-like attributes (such as IP addresses, port numbers, and flowIDs) that if they are not filtered out would enable the models to obtain high test-set accuracy by simply memorizing source and destination pairs, not learning the attack patterns. The behavioral finding may be trusted because the explicit leakage removal step (Table V, row 3) was incorporated in the preprocessing pipeline: the top features in Fig. 8 are truly behavioural, as identifier-like fields were removed from the feature space prior to training. This design choice, based on the well-known ML best practices [1] and [30] is key for any IDS model being deployed to traffic from different devices or network segments that have not been seen before.

C. Ablation Study

The pipeline design decisions are listed in Table XV and are used to provide a structured ablation analysis of the five key pipeline components. The most important is leakage field removal, as otherwise models could artificially memorize the association of certain IP addresses or flow IDs and the attack class, and would fail to recognize the attack class at near random, when presented with traffic from new network segments or devices [1], [30]. Balanced subsample class weighting is the second most powerful one for Random Forest. CSE-CIC-IDS2018 is class imbalanced, with the benign traffic as the majority class, and if not reweighted, the training loss of RF is dominated by the benign class, systematically under-penalizing attack misclassification. The recall of RF with weighting is 0.9165 while without weighting, recall would be much lower [16] because the weighting brings back the balance

TABLE XV. ABLATION STUDY CONTRIBUTION OF PIPELINE COMPONENTS

Pipeline Component	Impact if Removed	Affected Metric	Justification
Leakage field removal (Flow IDs, IPs, timestamps)	Models memorize endpoints; accuracy inflated but does not generalize to unseen traffic	Generalizability	Forces behavioral learning models detect traffic patterns, not specific sources [1], [23]
Balanced subsample class weighting (RF only)	RF over-predicts benign class due to class imbalance; recall drops significantly	Recall, FN count	CSE-CIC-IDS2018 is imbalanced; class weighting restores balanced sensitivity [16]
StandardScaler feature normalization	Tree-based models unaffected; non-tree extensions (SVM, LR) would fail without standardization	Future extensibility	Ensures pipeline consistency for future non-tree model extensions [4], [5]
Binary relabeling (multiclass → binary)	Multiclass task increases complexity; minority attack families reduce performance on small classes	Task complexity, F1	Binary task is the primary focus; multiclass is a planned future extension [18]
Stratified split strategy	Non-stratified split risks class-proportion mismatch between train and test sets	Evaluation fairness	Ensures test set is representative of the full class distribution [6], [25]

in the training signal. StandardScaler normalization does not affect tree-based model predictions (tree splits are invariant to monotonic feature transformations) but makes the pipeline capable of being extended to distance-based or gradient-based methods [4], [5] in the future.

D. Theoretical Scalability and Deployment Considerations

A comparative scalability and deployment analysis of the three model configurations as implemented based on the known theoretical properties of the model configurations is presented in Table XVI. Since the trees in XGBoost are shallow (tree depth 6, see Fig. 4), it is more suitable for the deployment at the edge for IoT.

E. Limitations

A number of limitations of this study should be noted, each of which is considered in the future research roadmap (Table XVII). First, only binary classification is assessed, which means that the benign traffic is separated from attack traffic with no further distinction of the seven attack families in CSE-CIC-IDS2018. Multi-class classification would offer more granularity into the nature of the attack families an individual model is good at detecting or not, such as whether the model is good at detecting brute force attacks, whether or not it's good at detecting DoS attacks, whether or not botnet C2 traffic is a harder boundary to detect because it's lower volume and less distinct a flow signature [2]. Second, time for training, time for testing, and memory was not measured. Such metrics are essential to determine if IoT edge deployment is feasible, as resource-constrained devices like smart home gateways have limited RAM and computation budgets and may be unable to support the configurations tested here, specifically the 200-estimator RF and 300-estimator XGBoost ones [13]. Third, the evaluation relies on a static train-test split based on a fixed stratification, whereas real network traffic in which the proportion of benign and malicious traffic varies over time because of the appearance of new variants of attacks [21], [22], [23]. Fourth, the use of SHAP-based per-prediction attribution was not fully integrated into the evaluation pipeline, and only allowed for making claims about global feature importance, rather than instance-level explanations [7]. Fifth, since the experimental validation of the suggested approach is carried out using CSE-CIC-IDS2018, there is no empirical evidence of the generalizability of the entire approach across datasets yet. With respect to this issue, the updated analysis also takes into account independent published findings from Adewole et al. [18], who demonstrated great performance of the XGBoost classifier on the IoT-oriented CICIoT2023 dataset. Although this information confirms the appropriateness of choosing the classifier, it does not substitute for the direct verification of the proposed vulnerability-aware approach on a different dataset. Future research will involve the application of the entire approach to other IoT datasets.

TABLE XVI. SCALABILITY AND DEPLOYMENT ANALYSIS COMPARATIVE ASSESSMENT

Aspect	Random Forest	XGBoost	RF+XGBoost Ensemble
Training complexity	O(n·d·T·log n); parallelizable across trees	O(n·d·T·log n) sequential; subsampling reduces effective cost	Sum of RF + XGBoost training costs
Inference complexity	O(T·depth) per sample; parallelizable	O(T·depth) per sample; comparable to RF	Sum of both model inference costs
Memory footprint (relative)	Higher: 200 full-depth trees	Lower: 300 shallow trees (depth=6)	Highest: combines both model footprints
Scalability to large datasets	Good: bootstrap sampling limits per-tree memory	Excellent: subsampling + regularization	Good: inherits component model scalability
Edge / IoT deployment suitability	Moderate: 200 full-depth trees may exceed IoT RAM	Better: shallow trees reduce memory requirements [13]	Lowest: dual-model overhead unsuitable for constrained devices

IX. CONCLUSION AND FUTURE WORK

This paper proposed a vulnerability-aware ML framework for intrusion detection in smart-home-relevant IoT environments, and evaluated it on the CSE-CIC-IDS2018 [2] dataset. It was a workflow that examined Random Forest [4] and XGBoost [5] alone, as well as an ensemble of the two methods using soft voting (RF+XGBoost) on more than 1.3 million

TABLE XVII. STRUCTURES FUTURE RESEARCH DIRECTIONS AND ROADMAP

Research Direction	Methodology / Approach	Expected Contribution
SHAP per-prediction attribution	Integrate SHAP [7] into evaluation pipeline alongside RF MDI feature importance	Provides instance-level explanation; validates vulnerability-aware mapping with prediction-level evidence
Multi-class attack classification	Extend binary pipeline to 7-class classification matching CSE-CIC-IDS2018 attack families	Enables finer-grained detection; reveals which attack families are hardest to distinguish
Training time and memory profiling	Instrument training and inference pipelines with time and memory profilers	Provides deployment feasibility assessment for resource-constrained IoT devices [13]
Temporal evaluation protocols	Time-ordered splits, sliding-window cross-validation, concept-drift detection	Improves real-world validity; models adapt to evolving attack traffic [21], [22], [23]
IoT-specific dataset validation	Apply framework to CICIoT2023, Bot-IoT [8], N-BaIoT [24]	Strengthens ecological validity; tests generalization across IoT traffic environments
Lightweight model compression	Feature subset selection, tree pruning, knowledge distillation for edge deployment	Enables on-device IDS on resource-constrained IoT hardware [13], [19]

held-out test flows.

Among the evaluated configurations, XGBoost proved to be the superior overall model, (accuracy: 0.9822, precision: 0.9969, recall: 0.9139, F1-score: 0.9536, ROC-AUC: 0.9907 and only 760 False Positives, which is 19 times lesser than RF), confirming the central hypothesis and answering RQ1. The XGBoost algorithm performance is also in line with independently published results from an independent study on the CICIoT2023 dataset designed specifically for the IoT environment, where Adewole et al. [18] achieved 98.54% accuracy and 98.55% F1-score. This independent validation provides additional justification for the selection of XGBoost as the detector in the proposed scheme, but further testing of the full vulnerability-aware pipeline is required to prove generalisability. The soft-voting ensemble did not make any significant improvements over standalone XGBoost, which agrees with the theoretical results of Tama and Lim [6] as shown in Fig. 5. In addition to this comparative evaluation, the paper presented: theoretical support by original arrow diagrams (Fig. 1, Fig. 2, Fig. 3, Fig. 4, Fig. 5) containing schematic diagrams of the architecture of each model and of the vulnerability-to-detection chain; a table of IoT vulnerability characterization (Table II); a vulnerability-aware interpretation layer (Table VII and Table XIV + Fig. 17, Fig. 2); a detailed ablation study (Table XV); a scalability analysis (Table XVI); 19 figures (14 real and 5 original theory figures); and a structured future research roadmap (Table XVII). All results and figures are based only on the real experiment or accepted theory.

The structured future research roadmap is given in Table XVII.

The results confirm the conclusion that tree based ML models and particularly the XGBoost based models offer an effective, interpretable and practically defensible basis for ID on the modern flow based benchmarks. Their fusion with structured vulnerability-aware reasoning extends the utility by providing a security context for the model outputs to help practitioners reason and act upon them, in line with the overall trend of explainable and deployment-aware IoT IDS research [3], [9], [13], [18], [20], [33].

ACKNOWLEDGMENT

The authors extend their appreciation to the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia [KFU264430]. The authors would like to thank the anonymous reviewers for their insightful scholarly comments and suggestions, which improved the quality and clarity of the paper.

REFERENCES

[1] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, p. 102419, 2020.

[2] Canadian Institute for Cybersecurity and Communications Security Establishment, "CSE-CIC-IDS2018 on AWS dataset description," University of New Brunswick, 2018.

[3] I. H. Sarker, "Explainable AI for cybersecurity automation, intelligence and trustworthiness in digital twin: Methods, taxonomy, challenges and prospects," Edith Cowan University, 2024.

[4] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[5] 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*, pp. 785–794, 2016.

[6] B. A. Tama and S. Lim, "Ensemble Learning for Intrusion Detection Systems: A Systematic Mapping Study and Cross-Benchmark Evaluation," *Computer Science Review*, vol. 39, p. 100357, 2021.

[7] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[8] N. Moustafa and J. Slay, "The Evaluation of Network Anomaly Detection Systems: Statistical Analysis of the UNSW-NB15 Dataset," *International Journal of Information and Computer Security*, vol. 3, no. 4, pp. 321–339, 2016.

[9] A. Saied, M. Al-Ayyoub, Y. Jararweh, and E. Benkhelifa, "AI-Based Intrusion Detection Systems in the Internet of Things: A Survey and Taxonomy," *IEEE Internet of Things Journal*, vol. 11, no. 2, pp. 1058–1075, 2024.

[10] U. Ahmed *et al.*, "Explainable AI-Based Innovative Hybrid Ensemble Model for Intrusion Detection," *Journal of Cloud Computing*, vol. 13, art. no. 150, 2024.

[11] S. Yaras *et al.*, "IoT-Based Intrusion Detection System Using New Hybrid Deep Learning Algorithm," *Electronics*, vol. 13, no. 6, art. no. 1053, 2024.

[12] X. Chen, M. Liu, Z. Wang, and Y. Wang, "Explainable Deep Learning-Based Feature Selection and Intrusion Detection Method on the Internet of Things," *Sensors*, vol. 24, no. 16, art. no. 5223, 2024.

[13] A. Javed *et al.*, "Embedding Tree-Based Intrusion Detection System in Smart Thermostats for Enhanced IoT Security," *Sensors*, vol. 24, no. 22, art. no. 7320, 2024.

[14] M. Sarhan *et al.*, "Feature Extraction for Machine Learning-Based Intrusion Detection in IoT Networks," *Digital Communications and Networks*, vol. 10, pp. 205–216, 2024.

[15] F. Ullah *et al.*, "IDS-INT: Intrusion Detection System Using Transformer-Based Transfer Learning for Imbalanced Network Traffic," *Digital Communications and Networks*, vol. 10, pp. 190–204, 2024.

[16] M. B. Musthafa *et al.*, "Optimizing IoT Intrusion Detection Using Balanced Class Distribution, Feature Selection, and Ensemble Machine Learning Techniques," *Sensors*, vol. 24, art. no. 4293, 2024.

[17] A. Alabbadi and F. Bajaber, "An Intrusion Detection System over the IoT Data Streams Using Explainable Artificial Intelligence (XAI)," *Sensors*, vol. 25, no. 3, art. no. 847, 2025.

[18] K. S. Adewole, A. Jacobsson, and P. Davidsson, "Intrusion Detection Framework for Internet of Things with Rule Induction for Model Explanation," *Sensors*, vol. 25, no. 6, art. no. 1845, 2025.

[19] V. Holubenko, D. Gaspar, R. Leal, and P. Silva, "Autonomous Intrusion Detection for IoT: A Decentralized and Privacy Preserving Approach," *International Journal of Information Security*, vol. 24, art. no. 7, 2025.

[20] D. Krishnan, S. Singh, and V. Sugumaran, "Explainable AI for Zero-Day Attack Detection in IoT Networks Using Attention Fusion Model," *Discover Internet of Things*, vol. 5, art. no. 83, 2025.

[21] M. M. Rahman *et al.*, "A Survey on Intrusion Detection System in IoT Networks," *Internet of Things*, 2025.

[22] S. Sallam *et al.*, "Intrusion Detection on the Internet of Things: A Comprehensive Review and Gap Analysis Toward Real-Time, Lightweight, Adaptive, and Autonomous Security," *Cybersecurity*, vol. 7, no. 1, art. no. 16, 2026.

[23] Y. Sanjalawe *et al.*, "A Review of Artificial Intelligence-Based Intrusion Detection in Industrial Internet of Things," *Discover Internet of Things*, 2026.

[24] Y. Meidan *et al.*, "N-BaIoT: Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.

[25] C. Yin, Y. Zhu, J. Fei, and X. He, "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[26] R. Vinayakumar *et al.*, "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[27] I. Cvitic, D. Perakovic, M. Perisa, and B. Gupta, "Ensemble Machine Learning Approach for Classification of IoT Devices in Smart Home Environment," *International Journal of Machine Learning and Cybernetics*, vol. 12, pp. 3179–3202, 2021.

[28] M. A. Khan and K. Salah, "IoT Security: Review, Blockchain Solutions, and Open Challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018.

[29] H. Hindy *et al.*, "A Taxonomy of Network Threats and the Effect of Current Datasets on Intrusion Detection Systems," *IEEE Access*, vol. 8, pp. 104650–104675, 2020.

[30] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges," *Cybersecurity*, vol. 2, art. no. 20, 2019.

[31] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed., Springer, 2009.

[32] D. S. Berman, A. L. Buczak, J. S. Chavis, and C. L. Corbett, "A Survey of Deep Learning Methods for Cyber Security," *Information*, vol. 10, no. 4, p. 122, 2019.

[33] A. B. Nassif, M. A. Talib, Q. Nasir, and F. M. Dakalbab, "Machine Learning for Anomaly Detection: A Systematic Review," *IEEE Access*, vol. 9, pp. 78658–78700, 2021.