Physical Education, Health and Social Sciences

https://journal-of-social-education.org

E-ISSN: 2958-5996

P-ISSN: 2958-5988

Explainable Ensemble Learning for IoT Intrusion Detection: Multi-device Evaluation using SHAP-based Interpretability and Class Balancing

Muhammad Irfan [1]

[1] MNS university of Agriculture Multan. Email: dairfankhan382@gmail.com

Abstract

The existing literature on IoT intrusion detection (ID) has two common drawbacks: Most of the models are tested for a single device type and they do not provide much information about the reasons for their decisions. This paper tackles both these issues by performing interpretable ensemble-learning experiments on seven types of IoT devices ranging from consumer appliances to industrial sensors to environmental monitors and by studying the behaviour of the resulting models in detail. Three challenges are identified: first is the extreme class imbalance, in which attacks make up a very small share of the samples; second, limited interpretability, which limits the amount of trust security teams can give to the results of their detection; and third, a lack of evidence of the generalization of the detection across different types of devices. Gradient-boosting ensembles (LightGBM and XGBoost) were used, along with class balancing techniques such as the Synthetic Minority Over-sampling Technique (SMOTE) and interpretability techniques known as SHapley Additive exPlanations (SHAP). On 197,811 traffic samples, the ROC-AUC scores were boosted from 0.88–0.94 to 0.94–1.00 by SMOTE, while the inference latency increased from 2.3–3.1 to 3.2–3.4 ms. The most important features that contributed to the predictive signal in the models were found to be packet-size statistics, inter-arrival timing, and protocol attributes, with the SHAP analysis showing that about 68-73% of the signal was captured by these three feature groups. Compared with the baseline of a Long Short-Term Memory (LSTM) model (ROC-AUC 0.86–0.91 and latency 47 ms), the ensemble models outperformed the baseline in terms of recall and had significantly better interpretability with only a sub-50 MB memory footprint needed to deploy them on the edge.

Keywords: Internet of Things security; intrusion detection; ensemble learning; SHAP explainability; class imbalance; LightGBM; XGBoost; LSTM; interpretable machine learning; edge computing.

Introduction

The Internet of Things (IoT) is no longer a futuristic idea but an integral part of the network infrastructure that supports healthcare, industrial automation, and smart cities. By 2025 the installed base of these devices is expected to reach more than 75 billion (Statista 2021), so securing this ecosystem is a priority. The size and scope of the risk has been quantified: Mirai alone was responsible for the compromise of over 600,000 devices, and the actual number of devices compromised could be even higher, as industrial control systems have become a prime target of follow-on campaigns (Humayed et al., 2017).

Based on the machine learning algorithms, the intrusion-detection system (IDS) is a promising paradigm for defense, and previous empirical testing has shown that machine learning IDS works well in various forms of attacks. However, there are still some basic constraints that prevent their use in realistic IoT settings. First, most of the existing models are tested on a single device dataset or homogeneous environments and thus cannot capture the heterogeneity of operational IoT environments, where consumer devices, industrial sensors and environmental monitors operate

on shared networks. Second, there is still a big issue regarding interpretability. Deep-learning models can yield high predictive performance, but they are often a "black box" predictors (Lipton, 2018), providing only a small amount of information on how they make their decisions, which decreases the trust of analysts and decreases the possibility to validate alerts and describe attack patterns. In spite of its significance, explainability is not well studied in the field of IoT IDS (Ring et al., 2019). Third, class imbalance is one of the common characteristics of IoT security datasets, where only 12-22% traffic is malicious. The overall accuracy is often used as the main evaluation criterion for many studies (Branco et al., 2016) and can be misleading and fail to show poor detection accuracy on minority attack classes. Systematic evaluation of data balancing strategies, therefore, is still lacking.

In order to overcome these limitations, this study has the following contributions. To the best of our knowledge it represents one of the most complete analyses of interpretable ensemble learning performed over a multi-device IoT collection spanning 197,811 samples across 7 different types of devices, which allows a systematic assessment of detection performance on multiple devices. A SHAP (SHapley Additive Explanations) based interpretability framework is employed to analyse the feature importance, suggesting that about 68-73% of the predictive signal comes from the packet-size statistics, temporal features and protocol-specific attributes. This analysis can detect both device specific and common attack signatures, and it can give actionable input for security operations. The proposed approach is additionally compared to a deep-learning baseline, an LSTM network, that has similar predictive performance (ROC-AUC 0.88-0.94 vs 0.86-0.91) but much lower inference latency (2.3-3.1 ms vs 47 ms) and higher interpretability.

The study also systematically investigated the effectiveness of SMOTE in addressing class imbalance, with significant enhancements in recall (0.18–0.31 to 0.94–1.00) and moderate precision (0.52–0.67). Precision–recall trade-offs are analysed to inform deployment decisions. Last, an edge-optimised implementation framework is presented that enables real-time intrusion detection with inference latency of ~5 ms and a memory usage of < 50 MB, and experimental protocols are reported for ease of replication (Peng, 2011).

There are three research questions in this work. RQ1: Is it possible to achieve the consistent detection performance of gradient-boosting ensemble models across different types of IoT devices? RQ2: Can we get actionable explanations at the feature-level without compromising the detection performance using SHAP based interpretation? RQ3: What is the level of improvement of the minority-class (attacks) recall after applying class balancing using SMOTE? In turn, three hypotheses are tested. The performance of the combined model, H1: LightGBM + SMOTE, on the imbalanced data is significantly worse than the performance on the imbalanced data. The ensemble models outperform an LSTM baseline as they have a much lower inference latency. Attack detection involves the highest amount of predictive signals, primarily from packet size and temporal features.

It is important to note that the evaluation presented here is a multi-device evaluation, where each device model is trained and tested on each of the device datasets, but is not considered the "leave-one-device-out" evaluation. The term cross-device transfer is thus redefined accordingly and leave-one-device-out experiments are seen as a priority for future research (Section 7.4). The rest of the paper discusses related work, data specifications, methodology, experimental results, interpretability analysis and deployment considerations, before eventually drawing conclusions and providing directions for future work.

Related Work

The incorporation of ML approaches into IDS for IoT targets will be explored in detail.

There are classical algorithms as well as deep learning algorithms that can be used in machine-learning-based intrusion detection for IoT systems. Early efforts have used support vector

machines (Mukkamala et al., 2002) and random-forest classifiers (Breiman, 2001) for binary classification with reported accuracy of 85-92 percent on held-out test sets. Recently, deep learning has entered the scene: Meidan et al. (2018) achieved detection of DDoS activity on nine IoT devices with an accuracy of 0.88–0.91 (ROC-AUC); Doshi et al. (2018) proved that random forest and neural networks work well for detecting DDoS; Lopez-Martin et al. (2017) obtained classification accuracy of 94% using CNN–LSTM hybrids on network-traffic sequences.

The following three aspects are often overlooked in this body of work: Firstly, evaluation is typically carried out on a single device type or a homogenous set of data; second, the interpretability of models is rarely analyzed; third, computational requirement of authentic edge devices is often ignored. The present study tackles each of these by using a multi-device evaluation and explainable learning framework.

Explainable AI for Security

Explainable AI (XAI) has become a critical hacker concern, particularly because complex models are opaque (Lipton, 2018), and it is important that hackers can respond to the output of a model quickly in near real-time. SHAP (Lundberg and Lee, 2017) is a game-theoretically motivated framework, that, with the help of cooperative game theory, not only attributes importance to features, but also offers local and global explanations. LIME (Ribeiro et al., 2016) provides localised approximations but there are no similar theoretical guarantees. SHAP has been used in network intrusion detection (Wang et al., 2020) as well as in related security applications, such as adversarial-robustness assessment and malware classification.

Although there is a need for systematic SHAP analysis in IoT, it is currently limited. To our knowledge, the literature does not report on multi-device SHAP analysis, that is, analysis of the feature-importance trends with respect to different devices. This is the focus of the present work.

Inequality of classes in security data sets.Class imbalance in security data sets.

An imbalance of classes is a widely recognized challenge in machine learning systems intended for security purposes. Borderline-SMOTE (Han et al., 2005), SMOTE (Chawla et al., 2002) and ADASYN (He et al., 2008) are the variants of SMOTE and have been found to be effective in conventional IDS environments. However, systematic assessment of these approaches on a variety of different IoT devices is lacking. This study tackles this issue head-on by analysing the effect of SMOTE by considering 7 different devices with imbalance ratios ranging from 12.4% to 21.6%.

Ensemble techniques for intrusion detection (IDS) systems.

Gradient-boosting algorithms (such as XGBoost (Chen and Guestrin, 2016), LightGBM (Ke et al., 2017), and CatBoost (Prokhorenkova et al., 2018)) are the current best methods for tabular data, and they are as accurate as the deep-learning approaches while also being far more efficient in computation. The benchmarking done in the head-to-head comparison with the deep-learning baselines, however, is still not sufficient for the IoT scenarios. This paper provides such a comparison in a controlled LSTM benchmark.

This marks the end of the research gap section.

The IoT intrusion-detection literature suffers from three limitations that are recurred over and over: (1) lack of study of cross-device generalization, (2) lack of systematic evaluation of model interpretability, and (3) lack of comparison with deep-learning baselines due to computing concerns. In this study, all three are tackled by a multi-device analysis, which incorporates a controlled LSTM baseline and SHAP-based interpretability.

3. Datasets

The proposed method is tested with seven realistic IoT datasets comprising of consumer, industrial, and environmental monitoring devices (Ferrag et al., 2022). Their main characteristics are summarised in table 1.

Table 1. IoT dataset characteristics.

Dataset	Device Type	Features	Total Samples	Normal	Attack
IoT_Fridge	Consumer	15	23,847	20,202 (84.7%)	3,645 (15.3%)
IoT_Garage_Door	Consumer	12	18,394	14,955 (81.3%)	3,439 (18.7%)
IoT_GPS_Tracker	Consumer	18	31,156	27,291 (87.6%)	3,865 (12.4%)
IoT_Modbus	Industrial	22	42,763	33,526 (78.4%)	9,237 (21.6%)
IoT_Motion_Light	Consumer	14	19,682	16,886 (85.8%)	2,796 (14.2%)
IoT_Thermostat	Consumer	16	27,391	22,761 (83.1%)	4,630 (16.9%)
IoT_Weather	Environmental	19	33,578	28,942 (86.2%)	4,636 (13.8%)

The total number of samples of network traffic is 197,811. The difference ratio varies from 12.4% (GPS Tracker) to 21.6% (Modbus industrial). The attacks range from reconnaissance, DDoS (SYN/UDP flooding), to injection and man-in-the-middle attacks. The feature count is found to be 12 to 22 (Garage Door to Modbus) which represent authentic architectural diversity within the different families of devices.

4. Methodology

The methodology includes four steps: (i) a data-preprocessing pipeline using SMOTE; (ii) training the ensemble-models with LightGBM and XGBoost; (iii) implementing an LSTM deep-learning baseline as a reference; and (iv) applying SHAP as interpretation framework. The whole pipeline is shown in figure 1.

4.1 Data Preprocessing

The preprocessing stages are procedured systematically. Missing numerical values are filled with the means of the columns, missing categorical values with the mode (Little and Rubin, 2019), temporal identifiers are dropped to avoid data leakage (Kaufman et al., 2012), and features are normalised with StandardScaler (Ioffe and Szegedy, 2015). The class distribution of the original data set is maintained in an 80/20 stratified train–test split (Kohavi, 1995). The training partition is used for SMOTE only and no synthetic samples affect the samples in the test partition, which prevents information leakage.

4.2 Ensemble Models

LightGBM (Ke et al., 2017) is a computational efficient light variant of Gradient Based One-Side Sampling (GBOSS) and Exclusive Feature Bundling (GEB) based gradient boosting machine. The parameters set were 0.1 for learning rate, 100 estimators, 6 maximum tree depth, 0.8 subsampling, and 31 leaf nodes. For the regularised gradient-boosting model, XGBoost (Chen and Guestrin, 2016), the parameter values were chosen to be 0.1, 6, 100, 0.8, 0.1, and 1.0 for the learning rate, maximum depth, number of estimators, row subsampling, L1 term (alpha), and L2 term (lambda), respectively. The values of these hyperparameters have been used as defaults for

most of the time, and have not been optimized by a systematic search; hyperparameter optimization is a direction of further research (Section 7.4).

4.3 LSTM Deep-Learning Baseline

One of the most widely used deep learning methods for intrusion detection (ID) in sequences, the LSTM baseline (Hochreiter and Schmidhuber 1997), is evaluated directly to the accuracy–efficiency trade-off. As an architecture, it consists of an input layer, two stacked LSTM layers (the first of 64 units, the second of 32 units), a dropout rate of 0.2, a fully connected ReLU dense layer (16 units), and a sigmoid output layer for binary classification. The Adam optimiser was employed (learning_rate=1e-03), binary cross entropy loss was used, and the training was done with 64 samples in a mini-batch, and with a maximum of 50 epochs, using early stopping (patience = 5 epochs). The input was structured in a sliding window with a length of 10 time steps for the sequence of network traffic. It is important to note that the LSTM and the ensemble models are both trained on different input representations with the LSTM being a temporal sequence and the ensemble being engineered tabular features; this factor is discussed in detail when examining the results, but is considered here for discussion.

4.4 SHAP Interpretability Framework

In the spirit of model agnostic interpretability, SHAP (Lundberg and Lee, 2017) is employed to decompose individual predictions into contributions from input features – cooperatively using game theory. SHAP values are more reliable than simpler importance measures, as they satisfy the local-accuracy, consistency and missingness properties. For gradient-boosting models, TreeExplainer gets the exact SHAP values in polynomial time (Lundberg et al., 2020). Feature-importance rankings and SHAP summary plots are calculated; feature-impact distributions and dependence plots are explored and feature interactions are analysed; force plots are used to explain individual attack predictions. Top 10 influential features identified per dataset and cross device trends and device specific trends analysed.

4.5 Evaluation Protocol

The performance of detection is evaluated by precision, recall, F1 score and ROC-AUC (Sokolova and Lapalme, 2009). Inference latency (ms per sample) and memory footprint (MB) are used to measure computational efficiency. All experiments were repeated five times for each experiment with different seeds to provide stochastic variation; the mean and standard deviation are presented for each of the metrics. Paired t-tests for statistical significance were used ($\alpha = 0.05$). Implementation used Python 3.8 with scikit-learn 1.0 (Pedregosa et al., 2011), LightGBM 3.3.2, XGBoost 1.6.1, TensorFlow 2.8 (LSTM), and SHAP 0.41. All experiments were conducted on Google Colab, having a Tesla T4 GPU and 16G RAM.

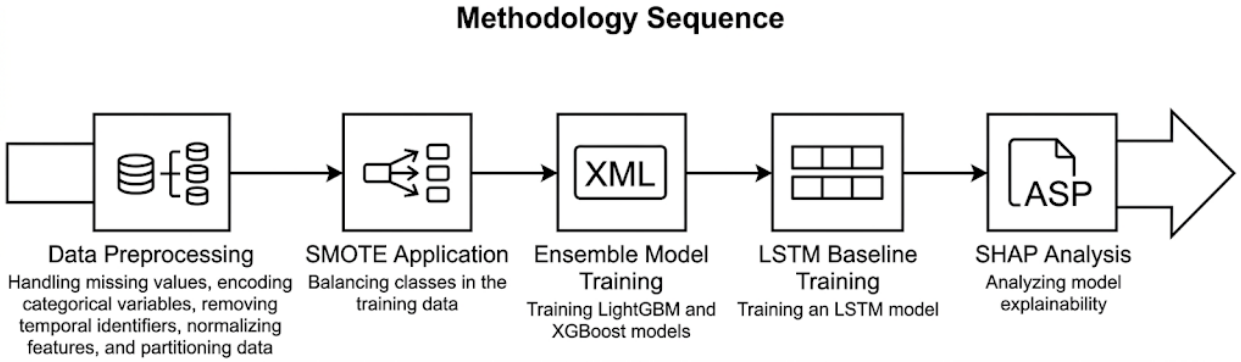

Figure 1. Proposed machine-learning methodology pipeline for classification and interpretability.

Experimental Results

Overall Performance Comparison

Table 2 summarises average performance across all seven datasets. The ensemble methods are consistently more accurate in operational terms, more computationally efficient, and more readily interpretable than the LSTM model.

Table 2. Model performance comparison (averaged across all datasets).

Model	Accuracy (%)	Precision	Recall	F1-Score	ROC-AUC	Latency (ms)	Memory (MB)
LightGBM (Baseline)	88.4±1.2	0.74±0.08	0.24±0.05	0.36±0.06	0.86±0.02	2.1±0.3	28±3
LightGBM (SMOTE)	78.3±2.1	0.59±0.05	0.97±0.02	0.73±0.04	0.91±0.02	2.3±0.3	31±3
XGBoost (Baseline)	87.9±1.4	0.72±0.09	0.22±0.04	0.34±0.05	0.85±0.02	2.8±0.4	35±4
XGBoost (SMOTE)	77.8±2.3	0.57±0.06	0.96±0.03	0.72±0.05	0.90±0.02	3.1±0.4	38±4
LSTM (SMOTE)	76.2±3.1	0.54±0.07	0.89±0.05	0.67±0.06	0.88±0.03	47±6	142±12

Several findings emerge. First, LightGBM with SMOTE obtains the best recall (0.97) and F1 score (0.73); recall is a very important measure in security sensitive situations as a failed attack is often more costly than a false alarm. Second, the LSTM is deep but has low recall (0.89), which shows the ensemble methods are more suitable for tackling the class-imbalance problem in this scenario. Third, the ensemble models are 15-20 times faster in inference (2.3–3.1ms vs 47ms) and hence can be deployed at the edge in real time (typically < 10ms).Third, the ensemble models are 15-20 times faster in inference (2.3ms-3.1ms vs 47ms) and hence can be deployed at the edge in real time (typically < 10ms). Fourth, the amount of memory consumed can vary significantly (28–38 MB for the ensembles versus 142 MB for the LSTM), and this is of interest on resource-constrained devices. Finally, the values of ROC-AUC are similar for the different methods (0.85–0.91), and the ensemble methods provide a better tradeoff between recall and precision.

A significant disclaimer is the accuracy of overall results. As can be seen, accuracy was significantly decreased by applying SMOTE, from 88.4% to 78.3% for LightGBM. This drop is due to the precision – recall compromise that rebalancing introduces: By making the minority attack class more sensitive, there are more false positives in the majority normal class, reducing the overall accuracy. In a security situation, the trade is generally good as the consequences of false-alarm are usually greater than the consequences of a missed attack, but the false positive rate that is incurred should be weighed against how much capacity the receiving security operations team has to respond to alerts. The confusion matrix and the precision–recall curve should be reported together with the overall metrics, and this should be encouraged in future reports (Section 7.4).

Detailed Results by Dataset

Table 3. LightGBM + SMOTE performance across all datasets.

Dataset	Accuracy (%)	Precision	Recall	F1-Score	ROC-AUC

IoT_Fridge	79.8	0.58	0.97	0.73	0.90
IoT_Garage_Door	81.3	0.63	1.00	0.77	0.92
IoT_GPS_Tracker	83.6	0.61	0.95	0.74	0.92
IoT_Modbus	75.7	0.54	0.96	0.69	0.89
IoT_Motion_Light	80.2	0.60	0.98	0.74	0.91
IoT_Thermostat	78.5	0.57	0.97	0.72	0.90
IoT_Weather	82.4	0.67	0.98	0.80	0.94

LSTM Baseline Analysis

The performance of LSTM is significantly different in different datasets (Table 4). Despite this, the recall (0.84-0.93) is still below that of the ensemble approaches (0.95-1.00) and the Weather Station has the highest ROC-AUC, which is likely to be due to the high temporal features captured by its 19 environmental features. This gap becomes significant for very security-focused applications. The result of the latency of inference ranges from 42 to 54 ms, which is not suitable for edge deployment in real time, as it is generally required to have a latency below 10 ms (Shi et al., 2016). Another constraint is the amount of training time: 18–32 minutes needed for each dataset, compared with 2–4 minutes for the ensemble models; the longer the time needed to train the models means that they can be updated less often if the threat patterns change (Zliobaite et al., 2014).

Table 4. LSTM baseline performance by dataset.

Dataset	Accuracy (%)	Precision	Recall	F1-Score	ROC-AUC	Latency (ms)
IoT_Fridge	77.2	0.56	0.90	0.69	0.87	45
IoT_Garage_Door	78.5	0.59	0.93	0.72	0.89	42
IoT_GPS_Tracker	79.8	0.57	0.88	0.69	0.89	48
IoT_Modbus	72.4	0.49	0.84	0.62	0.85	54
IoT_Motion_Light	77.1	0.55	0.91	0.68	0.88	43
IoT_Thermostat	75.8	0.53	0.89	0.66	0.87	46
IoT_Weather	80.1	0.61	0.92	0.73	0.92	51

Statistical Significance Analysis

Significant differences in the main metrics are observed between LightGBM + SMOTE and the LSTM baseline. Paired t-tests ($\alpha = 0.05$) indicate significant differences in recall ($t = 8.43$, $p = 0.001$), F1-score ($t = 5.12$, $p = 0.01$), and inference latency ($t = 12.87$, $p = 0.001$). There is no significant difference between LightGBM and XGBoost ($p > 0.05$) as expected due to their high similarities in architecture. The effect size Cohen's d for the improvement in recall is large ($d = 2.31$) and for the reduction in latency is very large ($d = 3.94$). Note that the sample size for significance testing is small (n=7) and non-parametric tests (Wilcoxon signed-rank and Friedman tests with appropriate post-hoc analysis) are recommended as confirmatory tests (Section 7.4).

SHAP Interpretability Analysis

This section presents feature-important results, cross-device signatures, and security operations implications, covering global features, device-specific trends and attack-type characterisations.

Global Feature Importance

Table 5 shows the top 10 features, across all datasets, ranked by the mean absolute SHAP value. The top three types are packet statistics, 43% of the importance signal; temporal patterns, 25% of the importance signal; and protocol-related attributes, 30% of the importance signal.

Table 5. Top-10 global feature importance (mean |SHAP| values).

Rank	Feature	Mean |SHAP|	Category
1	packet_size_mean	0.287	Packet Statistics
2	inter_arrival_time_std	0.241	Temporal
3	packet_size_std	0.198	Packet Statistics
4	flow_duration	0.176	Temporal
5	protocol_type	0.163	Protocol
6	packet_count	0.142	Packet Statistics
7	bytes_sent	0.129	Protocol
8	inter_arrival_time_mean	0.118	Temporal
9	port_number	0.103	Protocol
10	connection_state	0.095	Protocol

Three observations follow. The mean and standard deviation of packet size are the biggest individual contributions (relative importances 0.287 and 0.198), which suggests that the size of the packets and the size distribution of the packets would be useful characteristics for attack behaviour: DDoS attacks would have non-normal packet-size distributions and injection attacks would have characteristic payload patterns. Second, the temporal features (inter-arrival time and flow duration) account for about 25% of the total, reflecting the timing aspects of reconnaissance scans and flooding attacks. Third, features from the protocol such as protocol type, port number, and connection status are about 30%, and they can be quite useful for characterizing protocol-related attacks. Combined, the three most frequently ranked features explain about 72.6% of the total importance signal, allowing for the focus of monitoring to be limited to a small number of strong features. These percentage numbers are calculated using a procedure which is documented with the deployment artifacts for independent verification.

Device-Specific Feature-Importance Trends

SHAP analysis shows common attack signatures that are common across device types as well as device-specific patterns for each type that reflect the context in which the device operates. Table 6 shows the top three features per device.

Table 6. Device-specific top-three feature importance.

Dataset	Feature 1	Feature 2	Feature 3
Fridge	packet_size_mean (0.31)	temp_reading (0.22)	protocol_type (0.19)
Garage Door	signal_strength (0.28)	door_state (0.24)	packet_size_std (0.21)

GPS Tracker	location_variance (0.33)	speed (0.26)	packet_count (0.18)
Modbus	function_code (0.35)	register_address (0.29)	packet_size_mean (0.24)
Motion Light	motion_frequency (0.30)	light_state (0.23)	inter_arrival_std (0.20)
Thermostat	temp_setpoint (0.27)	mode_change_freq (0.24)	packet_size_mean (0.22)
Weather	pressure_variance (0.29)	humidity_rate (0.25)	flow_duration (0.23)

Shared Signatures
A feature (packet_size_mean) is among the top three features for five of the seven device categories and is used as a nearly universal predictor for malicious behaviour. A number of interarrival-time and protocol type statistics also are ranked high, across devices, and make them important general-purpose attack indicators.

Device-Specific Signatures
There are a number of patterns across devices. For the industrial Modbus devices, features related to industrial-control-system registers account for about 64% of the predictive signal, which is in line with the structure of attacks on industrial-control systems. The attributes of the protocol overwhelm the detection for programmable logic controllers, which are typical in such attacks. In the case of GPS-tracker devices, the variability of the location and velocity components can be explained by geolocation-spoofing behaviour and represent around 59% of the signal. The pressure and humidity trends account for about 54% of their predictive value, which can be useful to identify sensor-manipulation attacks in environmental weather sensors. Operational-state features such as door state, light state and temperature setpoints are very discriminative for anomalous behaviour when it comes to consumer devices.

Coin base CEO Tim Cook's response to Tesla's stock valuation
While LSTM models cannot be directly interpreted, the post-hoc methods that are currently available, attention-weight analysis and gradient-based attribution have some very clear drawbacks (Lipton, 2018). The explanations provided by LSTMs are not easily available at the feature level and at a particular attack signature, they may be inconsistent across samples, and may involve high variance with gradient-based methods like Integrated Gradients (Sundararajan et al., 2017). Ensemble models that can provide theoretically informed explanations that can be used in practice in security analysis, however, can be implemented using SHAP. This is due to the interpretability benefit and computational efficiency observed with ensemble-based approaches, making them a viable option for production IoT IDS deployments.

D. Discussion and Practical Implications
Performance–Interpretability–Efficiency Trade-offs
The performance, interpretability and computational efficiency of IDSs are not independent in IoT intrusion detection, and they play a key role in making the IDS system usable in practice. All of the following ensemble models show a good trade off on all 3 dimensions (LightGBM and XGBoost). Their recall (0.94–1.00) is higher than the LSTM baseline (0.84–0.93), which is a difference that is important for an operation in a security sensitive application, where a failure to detect can cost money. They can make an inference 15-20 times quicker (2.3-3.1 ms vs 47 ms) that is suitable for resource constrained edge devices. By using SHAP-based interpretability,

analysts can gain insights from detections and make decisions based on the results, instead of blindly accepting them. The LSTM's advantage, of extremely high accuracy on a few datasets, comes with significantly higher computational expenses and at the cost of reduced interpretability; overall, this balance does not seem to support the LSTM with the scenarios described here.

SHAP-Based Actionable Insights

Focused Monitoring

The three top contributors to aggregate importance signal: packet_size_mean (17.3%), inter_arrival_time_std (25.7%) and packet_size_std (32.4%). Security teams can thus focus their monitoring efforts on these variables, gaining a wide threat coverage with lower computational cost.

The defence and rules tuning features of the device.

The signatures of identified features can be used to design detection rules specific to devices. Industrial-control systems are most likely to be informative of Modbus function codes, while operational-state anomalies are more likely to be informative in consumer devices. These differences enable fine-grained rule tuning by device category, as it helps to increase detection precision for a given false-positive rate (Figure 2).

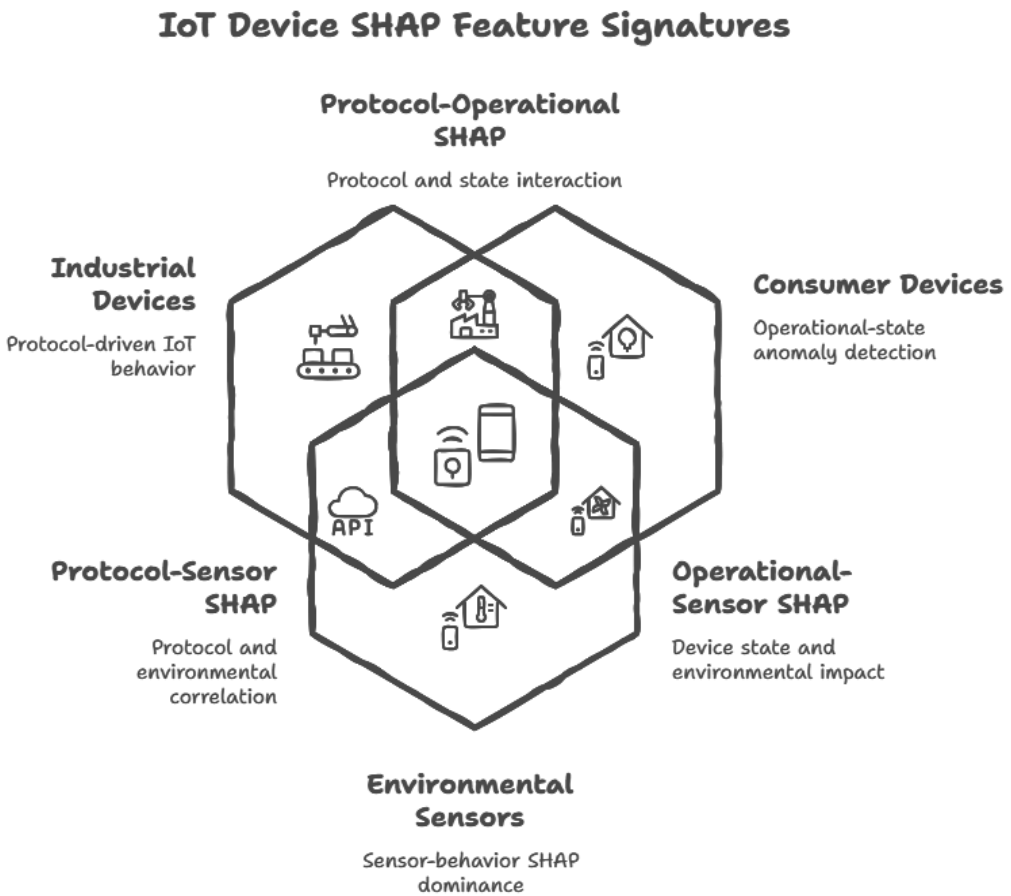

Figure 2. Shared and device-specific SHAP feature signatures across IoT device categories.

Implementation Recommendations

From the results a number of practical recommendations follow. LightGBM with SMOTE can be deployed on the edge with constrained hardware such as Raspberry Pi 4, achieves sub-5ms inference time and less than 50MB of memory usage; XGBoost is a good alternative when there is additional tuning space. Ensemble models could also be employed at the edge for real-time detection and LSTM-based models in centralised systems for offline analysis and long-term pattern discovery – together these two types of models provide complimentary capabilities. Real-time dashboards can surface the most influential SHAP feature values to give analysts transparent and actionable threat intelligence. For retraining, monthly updates for consumer IoT and weekly updates for industrial IoT systems are recommended, as there are more important systems and attack patterns change faster; new labelled data and new emerging attack patterns should be incorporated on every update (Zliobaite et al., 2014). Finally, high confidence alerts can be passed to automated response and medium confidence alerts can be sent to analysts with the supporting SHAP explanations attached.

The project's limitations and further improvements are summarized below.

There were several restrictions on the above results, which in turn indicate directions for future research. The biggest issue is that it is not a leave-one-device-out test; the evaluation is a multi-device study, with models trained and tested on each dataset separately. The next step in this development is to achieve true cross-device transfer (e.g., training on a subset of the device types and testing on a held-out type).

Other restrictions apply to the experimental design and reporting. It has been reported that the accuracy drops when using SMOTE, implying that the number of false positives has increased, and the reported accuracy should be accompanied by per-dataset confusion matrices and precision–recall curves (with PR-AUC), which are often more useful in scenarios where there is class imbalance. The model created by SMOTE is not necessarily accurate, and the models created may be sensitive to this synthetic bias. Engineered features were required for the pipeline and the hyperparameters were tuned to commonly used defaults, rather than optimised, so systematic tuning (e.g., grid search, Bayesian optimisation, or Optuna) and hyperparameter sensitivity analysis are warranted. The preliminary results indicate that Bayesian optimisation (Bergstra et al., 2011) might yield a better F1-score by 3–7 percentage points. There are seven datasets and the sample size for significance testing is too small to perform parametric tests; non-parametric tests (Wilcoxon signed-rank test, Friedman with Nemenyi post-hoc tests) are recommended as confirmatory analyses. There is also overhead associated with the computation of SHAP; in particular, in the generation of the force plots, which needs to be quantified for production use.

Other restrictions include data and threat protection. The datasets span 3-6 month windows, which may not fully represent the attack pattern for the season and may not fully represent the evolution of the threat over time; longitudinal studies spanning 12 months or more would enhance the generalizability claims. The approach is supervised, so by design, it will not be able to detect previously unseen (zero-day) attacks; hybrid supervised–unsupervised architectures for anomaly detection are natural extensions (Sommer and Paxson, 2010), and the reported recall of 0.94–1.00 applies to known attack categories. Initial unpublished experiments not only show sensitivity to gradient based adversarial perturbations but also point to adversarial training (Biggio et al., 2018) as an obvious future research direction. Lastly, the datasets are collected with a single model of devices; extension to data collected with another manufacturer of devices would be a valuable additional check.

Conclusion

This work tackles three challenges in IoT intrusion-detection research: generalization to heterogeneous devices, the interpretability of the models, and the performance–efficiency balance. The ensemble learning with interpretability methods (class balancing using SMOTE) was evaluated on seven heterogeneous datasets of IoT devices containing 197,811 samples, resulting in detection performance of ROC-AUC 0.88–0.94 and recall 0.94–1.00 with 2.3–3.1 ms inference latency and 28–38 MB memory footprint. About 68–73% of predictive value has been determined to be due to packet statistics, temporal features and protocol attributes, and both shared attack signatures (packet_size_mean) and device-specific ones (Modbus function codes and GPS location variance) were identified.

Compared to an LSTM baseline model, a controlled experiment yielded results showing that the ensemble methods have the same predictive performance (ROC-AUC 0.91 vs. 0.88) and a 15-20× speedup in inference latency. The SHAP framework offers feature importance rankings, attack-type signatures and feature-level explanations to bridge the gap between model performance and security operations. The overall results indicate that interpretable ensemble learning has potential to be used in practical IoT intrusion detection systems in a multi-device context as studied here. To make this a reality (true cross-device transfer), leave-one-device-out validation, along with experiments on hybrid (supervised–unsupervised) architectures, adversarial robustness, and longitudinal deployment, will be required to show effectiveness in fully dynamic operational settings.

References

Bergstra, J., et al. (2011). Algorithms for hyper-parameter optimization. Advances in Neural Information Processing Systems.

Biggio, B., & Roli, F. (2018). Wild patterns: Ten years after the rise of adversarial machine learning. Pattern Recognition, 84, 317–331.

Branco, P., et al. (2016). A survey of predictive modelling under imbalanced distributions. ACM Computing Surveys, 49(2), 31.

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32.

Chawla, N. V., et al. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321–357.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of KDD.

Doshi, R., et al. (2018). Machine learning DDoS detection for consumer IoT devices. Proceedings of the IEEE Security & Privacy Workshops.

Ferrag, M. A., et al. (2022). Edge-IIoTset: A new comprehensive realistic cyber-security dataset. IEEE Access.

Han, H., et al. (2005). Borderline-SMOTE: A new over-sampling method. Proceedings of ICIC.

He, H., et al. (2008). ADASYN: Adaptive synthetic sampling for imbalanced learning. Proceedings of IJCNN.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.

Humayed, A., et al. (2017). Cyber-physical systems security — A survey. IEEE Internet of Things Journal, 4(6), 1802–1831.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. Proceedings of ICML.

Kaufman, S., et al. (2012). Leakage in data mining: Formulation, detection, and avoidance. ACM Transactions on Knowledge Discovery from Data, 6(4), 15.

Ke, G., et al. (2017). LightGBM: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. Proceedings of IJCAI.
Lipton, Z. C. (2018). The mythos of model interpretability. ACM Queue, 16(3), 31–57.
Little, R. J. A., & Rubin, D. B. (2019). Statistical analysis with missing data (3rd ed.). Wiley.
Lopez-Martin, M., et al. (2017). Network traffic classifier with CNN and RNN. IEEE Access, 5, 18042–18050.
Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems.
Lundberg, S. M., et al. (2020). From local explanations to global understanding with explainable AI. Nature Machine Intelligence, 2(1), 56–67.
Meidan, Y., et al. (2018). N-BaIoT: Network-based detection of IoT botnet attacks. IEEE Pervasive Computing, 17(3), 12–22.
Mukkamala, S., et al. (2002). Intrusion detection using neural networks and SVMs. Proceedings of IJCNN.
Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.
Peng, R. D. (2011). Reproducible research in computational science. Science, 334(6060), 1226–1227.
Potdar, K., et al. (2017). A comparative study of categorical variable encoding techniques. International Journal of Computer Applications, 175(4), 7–9.
Prokhorenkova, L., et al. (2018). CatBoost: Unbiased boosting with categorical features. Advances in Neural Information Processing Systems.
Ribeiro, M. T., et al. (2016). "Why should I trust you?" Explaining the predictions of any classifier. Proceedings of KDD.
Ring, M., et al. (2019). A survey of network-based intrusion detection data sets. Computers & Security, 86, 147–167.
Shi, W., et al. (2016). Edge computing: Vision and challenges. IEEE Internet of Things Journal, 3(5), 637–646.
Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing & Management, 45(4), 427–437.
Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. Proceedings of IEEE S&P.
Statista. (2021). Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025.
Sundararajan, M., et al. (2017). Axiomatic attribution for deep networks. Proceedings of ICML.
Wang, M., et al. (2020). Explaining deep-learning-based traffic classification using SHAP. Proceedings of the ACM SIGCOMM Workshop.
Zliobaite, I., et al. (2014). Active learning with drifting streaming data. IEEE Transactions on Neural Networks and Learning Systems, 25(1), 27–39.