Diyala Journal of Engineering Sciences

Journal homepage: https://djes.info/index.php/djes

ISSN: 1999-8716 (Print); 2616-6909 (Online)

SHAP-Enhanced Histogram Gradient Boosting for IoT Threat Detection via Signal-Based Network Traffic Analysis

Ahmed A. Mohammed[1], Ahmed M. Aleesa[2*] and Ali M. Alhatim[1]

[1] Department of computer and information Engineering, Ninevah university, Mosul, Ninevah, 41002, Iraq
[2] Department of Programming, College of Computer Science and Information Technology, University of Kirkuk, 52001, Kirkuk, Iraq

ARTICLE INFO

Article history:
Received 01/07/2025.
Revised 08/09/2025.
Accepted 24/10/2025.
Available online 10/12/2025.

Keywords:
Histogram-based Gradient Boosting
IoT
Internet of Things
IDS
N-BaIoT

ABSTRACT

The dramatic rise in the number of Internet of Things (IoT) devices has greatly increased the size of the attack surface of network-based threats, especially high-volume, non-portable DDoS botnet attacks. Our hypothesis is to suggest an explainable intrusion detection system and analyse digital signals of raw IoT network traffic. We train a Histogram-based Gradient Boosting Classifier (HGBC) to identify benign and malicious traffic based on 11 classes (10 attack-related, 1 benign) on the N-BaIoT dataset. To reduce bias, the model has been trained on a strictly pre-processed and balanced subset of the data. We apply SHapley Additive exPlanations (SHAP), a game theory-based framework, to gain insight into complex model predictions that are security-relevant, despite the black-box nature of the model. This SHAP-enhanced method classifies and orders the most significant features, and it is found that mutual information and packet jitter characteristics descriptors (e.g., MI_dir_L0.1_mean) are decisive when identifying coordinated attack actions. The model reported the macro-averaged accuracy, recall and F1-score as 1.00 on a held-out test set. The three contributions of the work can be summarised as: (i) an end-to-end interpretable multi-class IoT DDoS detector; (ii) a transparent data curation framework that tackles imbalance and redundancy; and (iii) empirical support on how HGBC with SHAP can be highly performant yet offer actionable insight into the feature semantics that will inform future security design.

1. INTRODUCTION

The IoT technology is considered the new age technology. The interconnection between household items and industrial equipment changes the way societies live, work and interact with their surroundings. [1-3]. According to statistics in 2023, there are more than 15.9 billion IoT devices in the world, which clearly indicates that this technology is affecting almost every aspect of our lives. Nonetheless, the integration of IoT brings a series of security threats that call for well-established and efficient security countermeasures to protect these devices and the rich data they produce. Specifically, this paper sets out to discuss one such solution, which is utilising the Histogram Gradient Boosting Classifier for the identification of DDoS attacks. These attacks interfere with networks and devices connected to them and cause them to be unusable. [1, 2, 4-8]. IoT devices are resource-limited and are frequently placed in the open environment. Moreover, the rapid advancement and deployment of IoT devices have not been matched by adequate security measures. This has exposed them to threats that hackers exploit to launch severe DDoS attacks. The traditional methods of detection and identification prove to be inefficient for the current security needs. These approaches can be enhanced by using machine learning techniques; this, in turn, enhances the security. [9, 10].

Nevertheless, the threat environment has changed over the years, and IoT devices now perfectly match the DDoS attacks. These attacks can harm networks

[*] Corresponding author's Email: ahmed.marwan@uokirkuk.edu.iq

DOI: 10.24237/djes.2025.18404

and the devices that are connected to them with the aim of making them nonfunctional, thus creating a huge interruption [11, 12]. DDoS attacks exploit one or many of the characteristics which lead to this problem in the IoT application, including storage limitations and network capacity in the IoT devices. This has contributed to a tremendous rise in IoT-based botnet attacks that take advantage of weak security and complex, poorly configured protocols. According to the data, the IoT cyber-attacks of 2022 summed up to be at least 112 million [13]. This is an increase of 243 times, which is a daunting task for any candidate, regardless of his or her previous experience. This is a 39 percent improvement over the 32.7 million attacks registered in 2018. Also, as the scale of the total number of IoT devices expands, with over 15.9 billion connected IoT devices connected worldwide in 2023, and as this happens, so does the risk of cyber threats. Hence, there is a need to ensure that security features are enhanced in the IoT gadgets to reduce or eliminate the probability of these devices being compromised [14]. As many of these attacks are hard to detect and prevent, there are efficient machine learning methods to establish them, for example, Histogram Gradient Boosting Classifier. Being fast and accurate, this algorithm can be trained to look for signs of a DDoS attack and thus effectively prevent it [15-17].

The IoT technology is on the rise and is currently widespread in contemporary society. It has applications in many areas of life, such as health, automotive, energy, and homes. Overall, with the forecasted number of IoT devices being around 30 billion by 2030, this ecosystem can be characterised as highly prospective yet containing specific challenges [7, 18]. From the above risks, some of the malicious folks who want to take advantage of these vulnerabilities include botnet owners who have leveraged IoT to make large and distributed bot armies [19, 20]. These IoT botnets, including BASHLITE and Mirai, are very much active and hard to detect as they live on devices globally within the shadows. The Mirai attack of 2016 is a typical example to show the extent of devastation that old-style or new-style botnets are capable of [21]. In that incident, a massive DDoS attack was performed on Dyn, which disrupted access to some of the most popular sites, such as Twitter, Netflix, Reddit, and Spotify and proved that an IoT-driven DDoS attack can be deployed to trigger a large-scale disruption of services.[22, 23]. The impacts of such attacks are not only in the loss of services but also in privacy and personal security. [24]. Moreover, the smart IoT devices that people rely on can be infected without any obvious symptoms, and it is extremely hard to detect... There are many threats caused by the compromised devices, such as DDoS attacks, keylogging, stealing confidential data, and profiling operating systems. Due to such increasingly advanced dangers, it is high time to develop realistic IDSs. IDSs come in two primary forms: anomaly-based system, which includes anomaly detection, known as the identification of behaviour that is not normal and misuse-based detection systems, which are based on attack signatures. Using both concepts can be advantageous, but the approach based on anomaly detection tends to be more effective at finding new threats while being useless at telling the difference between them and false positives. Misuse detection, on the other hand, is able to detect well-known attacks but can barely detect new kinds of attacks [24-26].

The problem with security in such an environment is that the development of security measures should be able to adjust to the continuous permutation of strategies by the offenders. New directions in IoT security deploy machine learning and deep learning to detect threats much more effectively detect threats. Nevertheless, the use of outdated datasets has limited the existing approaches that do not paint an accurate picture of IoT attack scenarios in the present [27]. Several new datasets, such as N-BaIoT, have been created to address this gap. They consist of modern attack information that is collected by IoT devices containing malware. Using such datasets, it has been shown that the use of machine learning-based marking mechanisms enhances the capacity of the IoT environment to differentiate between the various attack features and therefore offer better protection [28]. Thus, it is vital to define the main characteristics of IoT systems, i.e. their heterogeneity and resource scarcity that distinguish them among the conventional networks. This is the reason why traditional security methods are not very efficient when it comes to IoT networks, as their devices are very complex. [29].

Therefore, this inclusive analysis of the problems associated with securing IoT reemphasises the need to expand the existing knowledge on threats as well as ideas on how a constantly growing IoT network can be defended. The future of IoT lies in our ability to protect its large network of interconnected devices against increasingly large botnet attacks. This study aims to enhance the ongoing discourse and strive to strengthen the protection mechanisms in the IoT environment, and continuously establish itself as a driver of change in the digital world [22, 30].

We represent raw IoT network flows as digital signals with the approach of previous studies, which have proven signal-based features to be useful in anomaly detection. An example is [31], which viewed IoT traffic as device signals to detect botnet traffic, and ProfilIoT, which used statistical decay-based features as indicators of the time-varying behaviour of IoT flows. A combination of these precedents prompts us to consider the following descriptors, namely, mutual information, packet entropy, and inter-arrival jitter, which are inherently signal-analytic and encode small deviations which are inherent to DDoS action. We have chosen the Histogram-based Gradient Boosting Classifier (HGBC) as the central learner, as its histogram-based binning greatly lowers the cost of training relative to classical gradient boosting, without compromising the accuracy of this model on high-dimensional traffic data. Additionally, HGBC facilitates L2 regularisation and early stopping, which reduce overfitting as a crucial importance in unbalanced IoT datasets, which reduces overfitting, a valuable factor in the imbalanced IoT dataset. Furthermore, HGBC is computationally cheaper compared to a wide range of deep learning counterparts, and therefore can be deployed in real-time or at the edge in IoT systems. In contrast to the earlier research, where the authors used boosting algorithms or explainability methods independently, this research proposes a closely connected framework, where HGBC is paired with the SHAP-based feature attribution. This combination is new to the IoT security field as it not only attains high multi-class classification with N-BaIoT traffic, but it also provides clear and security-relevant explanations of which features are driving detection.

In this regard, the N-BaIoT Dataset plays a significant role. [13, 32]. It offers a wealth of data in regards to IoT botnet attacks, which in turn can be used to improve the accuracy of the Histogram Gradient Boosting Classifier (HGBC). This work incorporates many attack scenarios, which makes the dataset useful to researchers and practitioners in IoT security. [13, 32]. Based on the Histogram Gradient Boosting Classifier (HGBC) and the N-BaIoT Dataset, it is quite probable to create a reliable detection system that could help to identify IoT DDoS attacks properly [33]. The proposed system can thus improve on the current security of IoT networks, closing gaps that expose the connected devices to external forces. Thus, this research seeks to evaluate the feasibility of using the Histogram Gradient Boosting Classifier (HGBC) algorithm in identifying IoT DDoS attacks by employing the N-BaIoT Dataset. Consequently, it aims at providing empirical evidence for the applicability of this approach during experimentation and at advancing current research in securing the IoT.

The remainder of this paper is organised as follows: Section two presents a comprehensive analysis of IoT DDoS attacks and their effect; Section three introduces Histogram Gradient Boosting and the rationale for using it for IoT DDoS attacks; Section four describes the N-BaIoT Dataset and how it can be useful to this study; Section five shows the experiment design and outcome; and lastly Section six summarises the paper and recommends future research.

2. SYSTEMATIC REVIEW OF THE RECENT LITERATURE

The latest research is still advancing the detection of IoT botnets. The strengths and limitations of ML/DL approaches are summarised using a 2023 systematic review of benchmark datasets and prep techniques. [34]. A 2024 taxonomy of AI-based DDoS detection techniques identifies as challenges in the field interpretability, detection in real time, and diversity of datasets [35]. One study suggests a SHAP-based federated learning architecture in 2025 and shows the usefulness of SHAP in decentralised and privacy-aware IoT intrusion detection[36]. Lastly, a 2024 study comparing NN models to N-BaIoT also reports inference-throughput experiments with edge hardware (Jetson Nano) and highlights the importance of scalable low-latency systems in IoT scenarios. [37]. In this section, we present the state-of-the-art work that serves as the basis of the current research undertaken to develop reliable machine learning models for IoT botnet detection. It is therefore imperative to have an understanding of the current existing literature in the domain of IoT security since the field witnesses rapid change and rising threats [37].

2.1 Machine Learning Paradigms for IoT Botnet Detection

The Internet of Things is quickly revolutionising the industries around the world by making them more connected and automated. Nonetheless, with this enhanced interconnectivity, cyber threats such as botnet intrusions, which are of great concern to IoT systems, can exploit a wider range of attack surfaces. Thus, recent literature has considered different machine learning models to address these threats, with much attention devoted to performance optimisation by using feature selection and dimensionality reduction.

One of the most important aspects of this process is dealing with the high dimensionality of data in the IoT network. Such methods as the Extreme Learning Auto-Encoder (ELAE) with applications by [38] Even though tested on standard image datasets, it emphasises a fundamental principle of the IoT: simplifying data without losing essential information can be a useful tool in the creation of effective detection models. Likewise, as observed by [39] In order to increase the detection performance of specific classifiers such as Multilayer Perceptron, Decision Tree, Support Vector Machine and Naive Bayes, it is important to optimise the models, and the

Grasshopper Optimisation Algorithm (GOA) was employed to achieve such optimisation. Their work highlights the importance of ideal feature and model selection as a critical step towards making a distinction between legitimate and malicious IoT traffic.

Author in [40] Demonstrated the trade-off between the dimensionality of the feature space and detection accuracy, a problem that is acutely felt by resource-constrained IoT operating conditions. The fact that they reduced their feature sets systematically on the DARPA data set with both SVMs and neural networks validated their claim that fine-tuning feature selection is the key to resource-efficient systems in intrusion detection. This was repeated by [41] using their minimal-redundancy-maximal-relevance (mRMR) feature selection and by [42], who introduced a mutual information-based feature selection to increase the performance of SVM in detecting botnet attacks.

Although these studies convincingly prove that feature selection and model optimisation are critical to attaining high accuracy and efficiency, they mostly consider the model as a black box. It is not about the reasons why features are semantically meaningful in security analysis, but about which features increase performance. They do not give any explainable security-relevant information on how certain traffic dynamics coded in these features are predictive of an active attack. This interpretability prevents a security analyst from knowing the nature of a threat and trusting the decision of the model to be effective in practical implementation.

This is where our work comes in to fill the gap. Although we also use sophisticated feature selection (through SHAP-based importance), we go beyond optimisation. In addition to an effective classification, the suggested SHAP-enhanced HGBC model can also present transparent, post-hoc explanations that assign predictions to particular traffic characteristics. This will enable security officers to know the logic behind each detection, e.g. detecting a sudden increase in mutual information (MI_dir_L0.1_mean) as a coordinated scan, thus closing the gap between high-performance classification and actionable security intelligence

2.2 Advanced Classification Strategies

In fact, IoT botnet detection is not limited to binary categorisations only. Authors in [43] extended the classification techniques to another level by integrating the Random Harmony Search algorithm for ranking features. This was complemented by a classifier based on Restricted Boltzmann Machines, specifically the Distributed Denial of Service (DDoS) kind. This is in line with their holistic nature of operation, which suits the complex categorisation of botnet attacks in IoT networking that may not easily fit into a straightforward categorisation. Furthermore, some of the key features which have been maximised include the number of neurons for LSTM and the number of neurons for the SLFN exhibit the complexity of IoT botnets. In straightforward ways, it is necessary to recognise different types of IoT botnet attacks to detect all of them efficiently. In [44] discussed the assessment of the Functional Trees classification technique, where the authors combined it with the concept of Genetic Search (GS). In their study, they assessed five various sorts of machine learning algorithms, which encompass J48, Naïve Bayes, Random Forest, Multilayer Perceptron, and Functional Tree. The authors' findings emphasise that accurate approaches for categorising influences distinctive botnet attack varieties within IoT networks. This capability is a very critical aspect in enhancing the security of IoT systems.

2.3 Unsupervised Learning with Autoencoders

Autoencoders, which are a type of deep learning, have been recognised as effective weapons when it comes to identifying IoT botnets. Authors [13] presented a novel method based on an unsupervised learning strategy. These techniques used by they is focused on capturing behavioural 'moments in time' of normal traffic inside IoT networks. Every single IoT device has its own autoencoder that helps introduce the device to the nature of the normal flow of traffic. Any deviation from these known patterns raises an alarm to the possibility of botnet activities. This approach shows a lot of potential for more efficient implementation in networks consisting of larger numbers of IoT devices, but can quickly be bogged down with the practicalities inherent in maintaining a unique model instance for each device.

2.4 Machine Learning for IoT Security

It is significant to note that ML and IoT security have evolved in interesting contributions jointly. For the more specific case of IoT-based networks, a network intrusion detection method was presented in Lopez-Martin et al. [45] based on the Conditional Variational Autoencoder (CVAE). Their innovation is to introduce intrusion labels into the decoder layers; in doing so, they eliminate certain levels of complexity linked to variational autoencoders. It is, however, recognised that such methods are not unique to IoT, although their applicability in industrial settings is quite apparent. Actually, while discussing the application of an effective IDS for IoT, the authors of [46] proposed a deep learning model of a feed-forward neural network tailored to IoT. Their experiments included binary and multi-class cases, and the performance of the models they built was rather high. This research aligns well with IoT security needs, where one is required to distinguish multiple botnet threats. Another work that has come to enrich the theme of anomaly-based Intrusion Detection Systems (IDS) within the IoT context was made by authors in [47], with their feature extraction and selection method. Their strategy built entropy-based information like Gain Ratio (GR) and Info Gain (IG) in order to identify and extract the features. This work offers some understanding of the best practices in today's world, where feature selection is crucial in IoT botnet detection. Authors in [48] conducted a comparative investigation on some of the deep learning methods, such as CNN and RNNs, especially the LSTM and the GRU networks. Their purpose was to detect anomalous, previously unseen behaviour within an IoT network, which could only occur within zero days, whilst keeping the FAR low. While not specific to IoT, the detection of zero-day anomalies is important at the best of times in industrial systems, where the failure to detect a botnet can have severe penalties.

2.5 Botnet Attack Vectors in IoT

This paper thus deems it essential to provide an understanding of botnet attacks in the IoT to lay a background for coming up with effective detection strategies. Among the most popular models is the client-server model, in which infected devices or hosts, often referred to as zombies or bots, report to a central server commonly known as the Command and Control (C&C) server. Instructions in this model are served from the C&C server to the zombie computers to facilitate the proclamation of various ill-natured activities.[49, 50]. However, as for the adaptation of the peer-to-peer model, the Botnet creates direct connections between the infected devices, leaving out the requirement of a server. While decentralised, this model poses unique challenges, as removing individual bots does not necessarily hinder the entire botnet. Distinguishing between these attack vectors is crucial for robust IoT botnet detection. [51] used the BLSTM-RNN detection model to detect botnets within consumer-based IoT devices and networks. Researchers used the word embedding technique to recognise text and convert attack packets into a tokenised integer format. They detected four attack vectors used by the Mirai botnet malware and evaluated them for loss and accuracy. According to the researchers of this study, the bidirectional technique added overhead to each epoch and increased processing time. However, it proved to be a better progressive model over time.

2.6 Detection Strategies

Researchers have devised a spectrum of strategies to bolster IoT botnet detection. Feature selection emerges as a critical facet in optimising intrusion detection systems. [40] Illustrated the potential for resource-efficient intrusion detection systems by systematically eliminating features and focusing on Support Vector Machines (SVM) and neural networks. This work highlights the intricate interplay between feature reduction and detection accuracy. Intrusion detection models have evolved beyond traditional paradigms. Authors in [43] Introduced the Random Harmony Search algorithm for feature selection and a classifier based on Restricted Boltzmann Machines for DDoS attack identification. Their comprehensive approach aligns with the multifaceted nature of botnet attacks in IoT. Furthermore, classification strategies tailored to differentiate between botnet attack vectors are essential, as demonstrated by Firdaus et al. [44] In their exploration of Functional Trees. The advent of deep learning techniques, particularly autoencoders, has ushered in new possibilities for unsupervised anomaly detection in IoT. Researchers in [13] proposed a novel approach based on behavioural snapshots of regular traffic facilitated by individual autoencoders for IoT devices. While promising, scalability concerns arise in more extensive IoT networks, where maintaining distinct models for each device could strain resources.

2.7 Synthesis and Insights

The literature reviewed helps build a background knowledge about the comprehensive strategies used to detect IoT botnets, which are directly similar to the actual informational background of our methodological design. The popularity of feature selection and dimensionality reduction tools [40-42, 47, 52] confirms our interest in the analysis of high-dimensional data of network traffic and proves the need to effectively represent features in any practical IoT security model. Moreover, the discussion of highly sophisticated methods of classification [43, 44] and sophisticated neural networks [13, 43, 45, 51] illustrates a strong direction in the field-wide approach to the multi-class, complex nature of current botnet attacks, and we have decided to stop focusing on binary classification. Nevertheless, an analytic review of this literature shows that three consistent, unremedied issues directly result in the research gap that our study will address:

The Interpretability Gap: Model interpretability has been sacrificed to the interest in maximising accuracy, which is dominant in the field. A large number of the most successful methods, including Deep Neural Networks (DNNs), Convolutional

Neural Networks (CNNs), Recurrent Neural Networks (RNNs such as LSTMs and GRUs), and ensemble methods [45, 46, 48, 53], are black-box methods. They attain high detection efficiency and do not give security operators a justifiable understanding of why a traffic event is considered an attack. A serious flaw in operational environments, this lack of transparency undermines trust and makes it more difficult to respond to incidents quickly and intelligently.

The Scalability Gap: Solutions put forward tend to ignore the high computational requirements of an actual IoT ecosystem. Techniques where training and a distinct model per device are required (e.g. the autoencoder-based method in [13], are inherently non-scalable in networks with millions of devices. Equally, high complexity models such as BLSTM-RNNs [51] They are computationally expensive and thus are not suitable for detecting in real-time, on resource-constrained hardware.

The Granularity Gap: A large part of the literature is limited to binary classification (benign vs. malicious)[52, 55, 60, 61]. This does not provide the actionable granulometry needed by security teams that need to determine the exact type of attack (e.g. Scan, UDP Flood, TCP Flood) to initiate a corresponding and effective mitigation response.

Table 1: Prior Work on N-BaIoT IoT Botnet Detection (2020–2025)

Ref	Year	Dataset	Task	Method	XAI / Interpretability	Best-reported result (as stated)	Limitations (as reported)
[54]	2021	N-BaIoT	Binary & Multi-class	MI-based feature selection + ML	No	99.9% (KNN with MI)	Limited interpretability; computational cost not discussed
[55]	2021	N-BaIoT	Binary	FS + LR / ANN	No	99.98% (LR+ANN)	Interpretability not addressed
[52]	2020	N-BaIoT	Binary	Fisher Score + XGBoost	No	99.96%	No interpretability; no scalability analysis
[53]	2022	N-BaIoT	Multi-class	XGB-RF hybrid	No	99.94%	Interpretability missing; dataset imbalance not addressed
[30]	2025	N-BaIoT	Binary	RF + SHAP, LIME, Rule distillation	Yes	High accuracy (reported near 99%)	Notes dataset bias; highlights need for explainability
[36]	2025	N-BaIoT	Binary	Federated learning + SHAP distillation	Yes	Competitive accuracy; 99.99%	Communication overhead in FL; still early-stage
[56]	2025	N-BaIoT	Binary & multi-class	VAE-GCN, VAE-GAT, VAE-MLP, and ViT-MLP	No	86.42%, 89.46%, 99.72%, and 98.38%	Computationally expensive; interpretability is not the focus
[57]	2025	N-BaIoT	multi-class	Attention-CNN-BiLSTM	No	~99% accuracy	Inference throughput not analysed; no XAI
[58]	2024	N-BaIoT	multi-class	Federated learning + SHAP aggregation	Yes	~99%	Focus on FL explainability; not tested in real deployment
[59]	2025	N-BaIoT	Binary	Federated Averaging (FedAvg)	No	97.5%	Focused only on privacy; no interpretability
[60]	2024	N-BaIoT	Binary	Dimensionality reduction + deep autoencoder	No	Good anomaly detection accuracy 90-99%	No interpretability; scalability not analysed

Addressing the Proposed Methodology: The following challenges were directly used to design our SHAP-enhanced Histogram Gradient Boosting (HGBC) framework. In order to overcome the interpretability gap, we incorporated SHAP (SHapley Additive exPlanations), a single model interpretation framework. This can enable our model not only to make predictions in discrete categories, but also to give specific explanations at a feature level about each prediction, the exact dynamics of the traffic (e.g. a spike in MI_dir_L0.1_mean) indicating the presence of an attack. To address the issue of the scaling gap, we chose the HGBC algorithm due to its established computational effectiveness, as well as the fact that it can work with large-scale datasets. We have a single, unified classifier across all 11 classes, and the model is much more scalable than the per-device models. Lastly, our approach fills the granularity gap by explicitly modelling and assessing it on an 11-class complex classification problem that offers the fine-grained threat detection required in real security operations. In conclusion, our proposed method is a direct response to the unresolved limitations within the current state-of-the-art, unifying high performance with the explainability, efficiency, and granularity required for deployable IoT security. We have summarised some of the most important recent studies that have used machine learning methods to detect IoT botnets based on the N-BaIoT dataset in Table 2. We have concentrated on publications within the past five years, with more recent publications (2024-2025) given priority to compare our work with the latest developments in the area. The table shows the dataset of each study, the methodology approach, the consideration of the interpretability, the best reported results, and limitations as reported by the respective authors. This systematic literature review can enable us to (i) stress the fact that a limited number of explainable AI applications in disrupting IoT botnets is easy to notice, and (ii) highlight that even the methods with very high accuracy lack analysis of generalisation, computational cost, and feature interpretability gaps that our SHAP-powered HGBC model attempts to offer.

3. PROPOSED METHOD

This section provides an overview of the materials used, data sources, the methods and approaches we used in conducting our research to arrive at a Reliable Machine Learning Model for IoT Botnet Detection. It comprises a dataset, dataset pre-processing, a feature selection method and the machine learning algorithms. A systematic methodological flow was developed to present a clear description of the research procedure. It starts with the description of the datasets and the definition of possible bias and feature relevancy, then continues with the dataset preparation phase: acquiring, balancing and preprocessing the dataset. A feature selection technique is then used to narrow down the input space before model development is carried out with the Histogram Gradient Boosting classifier.
Boosting Classifier. The last step in the methodology is the results evaluation and a critical analysis of results based on SHAP (to guarantee the interpretability). The chronological order of these processes is shown in Figure 1.

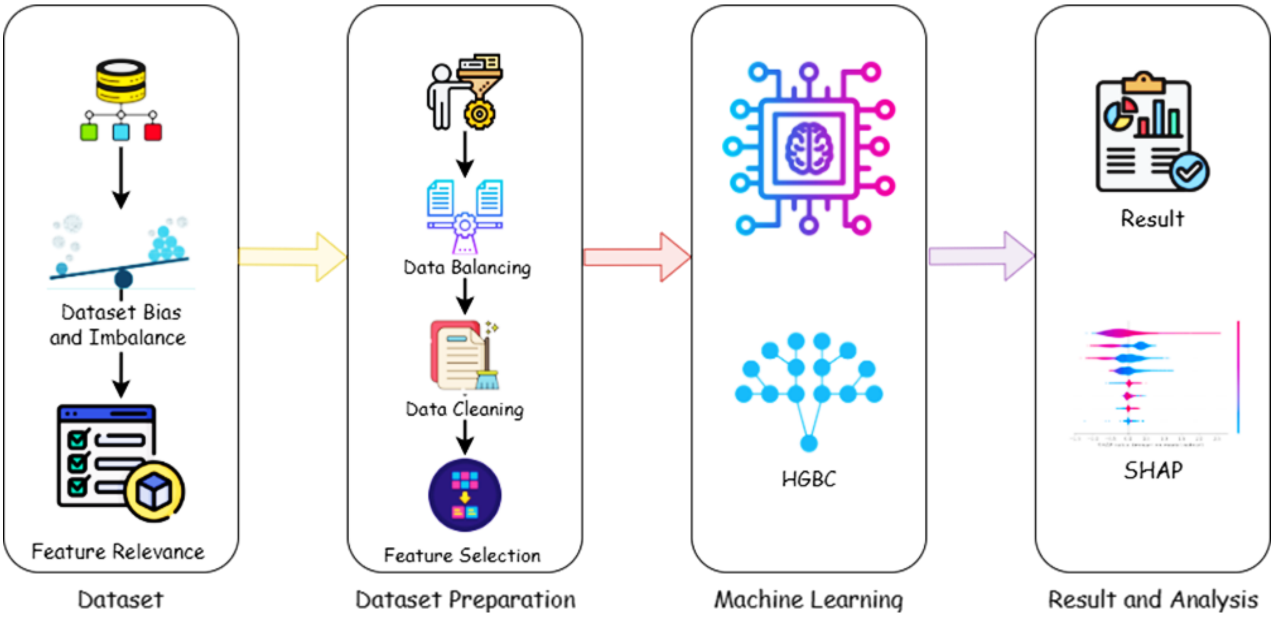


Figure 1. Flowchart of the proposed research methodology

3.1 Dataset Description

N-BaIoT is a new and advanced data collection intended to contribute to research in the field of IoT security. This dataset provides a thorough understanding of IoT device behaviour in a realistic environment because it is built from millions of records of real network traffic. The data is extremely complex and consists of 7,062,604 entries of malicious and benign network traffic, all simulated in a controlled environment. It includes nine different IoT devices that were attacked by two well-known botnets: Mirai and BASHLITE [13, 62-64]. These botnets were selected due to their background in being used to target IoT-based devices that have low security. The dataset consists of 5 variants of BASHLITE attacks (Scan, Junk, TCP Flood, UDP

Flood, COMBO), 5 variants of Mirai attack (Scan, Ack, Syn, UDP Flood, UDP Plain), and is described in Table 1 below.

3.1.1 Addressing Dataset Bias and Imbalance

Critical review of the original N-BaIoT dataset indicates that there is a severe inherent bias: an extreme imbalance between classes. As Table 3 indicates, some types of attacks are significantly overrepresented. The imbalanced nature of this raw data would greatly skew a model trained on this data to the majority classes, which is a well-known issue in machine learning and reduces the quality of the model when used on the minority classes [65]. To reduce this bias, we adopted an explicit balancing method by using random under sampling, which is a well-known approach to controlling class imbalance [66]. This will guarantee the classes are equally weighted when training the model and when evaluating it, resulting in a stronger and more accurate multi-class classifier. It should be mentioned that although this step is essential to methodological rigour, it implies that the model is trained on a curated dataset. Thus, its applicability to out-of-balance-sheet, real-world traffic would have to be validated in future research by testing on other independent datasets of IoT, which is a common method of testing the use of ML-based IDS [13].

These signal-based statistical features have been selected based on the previous research [31] that proved their effectiveness in describing the behavioural fingerprint of an IoT device and its anomalies during an attack. The core 23 features are used to define basic traffic characteristics such as packet timing, size and direction and their statistical interactions. To offer insight into these categories of core statistical features, Table 2 outlines those categories and gives an example of the corresponding expanded features that proved to be the most substantial in our analysis, and how we might interpret them in terms of security.

To provide an example, a burst in packet count is a well-known signature of a flooding-based DDoS attack [67], whereas mutual information (MI) features may be used to identify coordinated command-and-control traffic[10, 24]. The critical role of these particular types of features in detecting threats in IoT is also quantitatively validated in our subsequent SHAP analysis (Section 5). These signal-based statistical features are selected based on previous studies [13, 31] This showed that they could reveal the behavioural fingerprint of IoT devices and the variations during attack scenarios. The critical significance of these specific types of features in the detection of IoT threats is then quantitatively justified in our following SHAP analysis.

Table 2: List of attack types in the N-BaIoT dataset

botnets	Attack No.	Attack Type	Description
BASHLITE Attacks	1	Scan	Scanning the network for vulnerable devices
	2	Junk	Sending spam data
	3	UDP	UDP flooding
	4	TCP	TCP flooding
	5	COMBO	Sending spam data and opening a connection to a specified IP address and port
Mirai Attacks	1	Scan	Automatic scanning for vulnerable devices
	2	ACK	Ack flooding
	3	SYN	Syn flooding
	4	UDP	UDP flooding
	5	TCP	TCP flooding
	5	UDPplain	UDP flooding with fewer options, optimised for higher PPS

3.2 Dataset Preparation

Preparation of the dataset is a key aspect of the machine learning pipeline to improve the quality of data and its model resilience. To build a balanced, clean and normalised data set that we could use to train the Histogram Gradient Boosting Classifier (HGBC), we had to go through a few steps.

3.2.1 Data Acquisition and Balancing

This was initiated by obtaining data from the nine IoT devices of the N-BaIoT dataset, which includes Mirai and BASHLITE botnet traffic [13]. The initial data, presented in Table 2, experiences a severe class imbalance, with certain types of attacks (including mirai.udp) being much more prevalent than some others (including gafgyt.junk). A model trained on this raw data would be biased towards the majority classes, which is a well-known issue that drastically reduces the performance of minority classes detection [65].

We built a balanced dataset in order to reduce this bias and allow a fair assessment of all classes. The size of 30,000 samples per class was chosen on the basis of three main criteria. First, once the duplicate packets had been eliminated, the smallest minority classes (e.g., gafgyt.junk and gafgyt.scan) had about 31,000 unique instances. This size was determined to use the highest number of valid, non-synthetic data of all classes that would retain data integrity and prevent artefacts that may be caused by oversampling methods [68]. Second, the sample size is statistically large enough to surpass typical heuristics on the size of the minimum viable sample in machine learning (e.g., >10,000 samples [68] and offer a large amount of data to the model to learn the feature distributions of each attack type without being misled by imbalanced priors. Third, maximising Data Integrity by fixing this size will enable us to use 100 percent of the available unique data of the smallest minority classes (approximately 31,000 instances) after the elimination of the duplicates. This puts more emphasis on using real and non-synthetic data to prevent possible artefacts of oversampling methods and maintain the naturalness of the dataset[66]. The method of Random Undersampling of the very large majority classes [68]. Although such balancing gives priority to high per-class accuracy and excellent multi-class performance, it does so at a price: some data on the large majority of classes is lost. The final balanced data set was made of 330,000 samples (30,000 samples in 11 classes).

3.2.2 Data Cleaning and Preprocessing

The data analysis was done strictly in order to provide integrity.

- **Missing values:** no missing values were identified in the N-BaIoT dataset in its 115 features; consequently, no imputation was necessary.
- **Duplicate Removal:** To avoid overfitting the model and inflating metrics of performance, duplicate feature vectors have been eliminated. This procedure decreased the overall dataset size of 7,062,617 to 2,482,685 instances before balancing, as in Table 3.
- **Normalisation:** The HGBC algorithm requires normalised data to perform well [69]. As a result, all the numbers were normalised with Standard Score (Z-score) Normalisation. The same transformation was performed on the training set, and then the set was transformed with the parameters (mu, sigma) so that data leakage could be avoided. z is the transformed feature of a value x and is given by:

$$z = \frac{(x - \mu)}{\sigma}$$

Where μ = mean and σ=standard deviation of the feature, computed on the training set and used on the test set to avoid data leakage.

Train-Test Split: Train-Test Split: Following preprocessing, the balanced dataset was divided into a training set and a held-out test set using an 80/20 stratified split. Stratification keeps the class distribution consistent between the two subsets, and the ratio is a common metric that provides a potent gauge of generalisation performance. [70].

Table 3: Size and Number of packets in the used dataset Before\After Removing Duplicates

Label	Malware	Attack types	Before Removing Duplicates		After Removing Duplicates	
			No. of packets	File size	No. of packets	File size
0		benign	555,933	1.12 GB	513,498	1.006 GB
1	Bashlite	gafgyt.combo	515,157	1.04 GB	62,214	122 MB
2		gafgyt.junk	261,790	528 MB	31,294	59.7 MB
3		gafgyt.scan	255,112	496 MB	31,088	57.2 MB
4		gafgyt.tcp	859,851	793 MB	97,031	88 MB
5		gafgyt.udp	946,367	932 MB	107,666	105 MB
6	Mirai	mirai.ack	643,822	495 MB	280,145	236 MB
7		mirai.scan	537,980	298 MB	256,152	174 MB
8		mirai.syn	733,300	561 MB	317,115	267 MB
9		mirai.udp	1,230,000	1.53 GB	555,974	696 MB
10		mirai.udpplain	523,305	427 MB	230,508	198 MB
	Total		7,062,617	8.22 GB	2,482,685	3.0089 GB

3.3 Feature Selection

The selection of features is an important factor in developing a functional and meaningful intrusion detection model. It seeks to determine the most relevant input variables to predict the target class and hence minimise the computational complexity, as well as minimise noise. In this paper, the importance of the features was estimated with the help of a Decision Tree classifier, a powerful algorithm to rank the features by their potential to enhance the purity of the node (Gini impurity) in all possible splits of the tree [55, 71]. The importance scores were used to select the best features to be used to train the final Histogram-Based Gradient Boosting Classifier (HGBC) model.

Two main factors which led to the adoption of a tree-based method of feature importance were the model compatibility and computational efficiency. This method offers an effective feature ranking as a straightforward by-product of training a single, simple model, at minimal overhead with respect to more complex wrapper algorithms, like Recursive Feature Elimination, which involve training many models [72, 73]. Moreover, the features which a Decision Tree has identified as important are naturally well-adapted to the further tree-based ensemble models such as HGBC, since the two algorithms focus on the splits that will produce the maximum amount of information and purity of nodes [69].

The top 20 features that are discovered through this process are shown below. These features are no longer abstract statistical objects but have direct interpretable meaning to the network security analysis in that they are obtained as a product of Mutual Information (MI), Jitter, Covariance and Pearson Correlation (pcc) of network dynamics, which are proven indicators of malicious activities in the network[31].

MI_dir_L0.01_mean, MI_dir_L0.01_variance, HH_jit_L5_mean, HpHp_L3_covariance, HH_L3_covariance, HH_L0.01_covariance, HH_L5_covariance, HH_L0.1_covariance, MI_dir_L0.1_mean, HpHp_L5_covariance, MI_dir_L0.1_weight, HpHp_L0.01_covariance, MI_dir_L0.1_variance, HH_L1_pcc, HH_L1_covariance, HH_jit_L3_mean, HpHp_L5_mean, HpHp_L1_pcc, HpHp_L5_pcc, HpHp_L0.1_covariance.

The fact that these particular categories were the most salient indicates that the model is concerned with the important attack signatures:

- Mutual Information: (e.g., MI_dir L0.01 mean, MI_dir L0.1 mean): Large values are the result of predictable, synchronised patterns of two-way communication, a powerful signature of coordinated botnet command-and-control (C&C) behaviour or scanning patterns [67].
- Jitter (e.g. HH jit L5_mean, HH jit L3 mean): This is a measure of the variation of the packet delay. Attack traffic (e.g. UDP/TCP floods) is very erratic and bursty, and therefore produces much greater jitter than the benign traffic flow [13].
- Covariance (e.g., HH_L5_covariance, HpHp_L0.01_covariance): large values of covariance between the timing measurements of packets indicate non-linear relationships that are difficult to capture with the simpler metrics [13].
- Where HH L1 pcc, HpHp L5 pcc are Pearson Correlation: Pearson Correlation quantifies linear correlation between metrics, which can be used to detect particular coordinated attack vectors[13].

The choices of these characteristics confirm that the detection capabilities of our model are motivated by semantically meaningful changes in traffic based on coordination, timing instability, and intricate interactions between packets, which are core attributes of IoT botnet attacks. This interpretability is also confirmed and examined by the SHAP analysis in Section 5.

3.4 Machine Learning Model

We use the Histogram-Based Gradient Boosting Classifier (HGBC), which is an effective ensemble learning model that is uniquely used in massive data sets and feature space dimensions that are typical of an IoT network traffic analysis.

3.4.1 Histogram Gradient Boosting Classifier

The HGBC algorithm is particularly appropriate for real-time IoT security applications because it achieves impressive computational efficiency and high predictive accuracy by combining the concepts of gradient boosting with histogram-based approximation techniques, representing continuous features as discrete ones to speed up training and to save memory [74]. In contrast to the classical gradient boosting approaches that consider all possible points at which features may be split, HGBC groups feature values into a small number of bins (generally 255 by default), which significantly simplifies the computational effort required to determine optimal splits when constructing a tree. It is inspired by LightGBM [75], and has been shown to perform state-of-the-art on many classification problems, such as medical diagnosis, pavement condition assessment, and bioinformatics[76].

In our multi-class classification task (11-class IoT traffic classification), the HGBC constructs a single tree per class at a time, and optimises the model sequentially to minimise the multinomial log-loss (cross-entropy) function.

The HGBC is written by optimising a specialised objective function, the gradient and the Hessian of a loss function. The gradient shows in which direction the model predictions have to be changed, and the Hessian shows the shape of the error surface. The components steer the algorithm to narrow down its predictions by adding trees in an iterative fashion [74, 75]. The optimisation process is repeated in a series of calculating gradients and Hessians of individual points and then constructing histograms to calculate the gradient and Hessian histogram of the whole dataset. The prediction errors in the model are reduced due to the fact that fitting a new tree to residual errors reduces the errors in the prediction by the ensemble [75].

The mathematical formulation of the HGBC is a gradient-boosting ensemble whose base learners are decision trees: at each boosting iteration, a new tree is fitted to reduce the ensemble's residual error. Formally, after t iterations, the model prediction for instance x_i is:

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) \qquad (1)$$

Every$f_k \in F$ is a regression tree. The empirical objective that is regularised and minimised using boosting is:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{N} \ell\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{k=1}^{t} \Omega(f_k) \qquad (2)$$

Where ℓ (·,·) is loss (and we use categorical cross-entropy since there are 11 classes) and $\Omega(f)$ is a tree complexity penalty (as leaf-weight regularisation). To build trees, boosting fits the negative gradient of each new tree (and may additionally make use of second-order information). Indicating loss derivatives first and second at iteration t by:

$$g_i^{(t)} = \frac{\partial \ell(y_i, \hat{y}_i)}{\partial \hat{y}_i}\Big|_{\hat{y}_i = \hat{y}_i^{(t-1)}} \qquad (3)$$

$$h_i^{(t)} = \frac{\partial^2 \ell(y_i, \hat{y}_i)}{\partial \hat{y}_i^2}\Big|_{\hat{y}_i = \hat{y}_i^{(t-1)}} \qquad (4)$$

The optimum leaf weight ω^* for a node that wraps instances i with a second-order approximation is:

$$\omega^* = -\frac{\sum_{i \in I} g_i}{\sum_{i \in I} h_i + \lambda} \qquad (5)$$

In which λ is an L2 regularisation of leaf weights. The gain of a candidate split, which splits node I into I_L and $I_{R,}$ may be expressed as:

$$Gain = \frac{1}{2}\left(\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}\right) - \gamma \qquad (6)$$

Where $G_* = \sum_{i \in I*} gi$, $H_* = \sum_{i \in I*} hi$, and γ represents the penalty of creating a leaf. The following equations (1-6) are based on the standard derivation of gradient boosting [77] and are the basis of the current implementations. The histogram variant speeds up finding splits by quantizing continuous features into a fixed size bin set and summing g and h g and h within a bin (histograms The algorithm scans histogram bins (aggregated gradients/hessians), instead of scanning sorted continuous values, significantly reducing memory traffic and split-search cost with only a small difference in predictive performance to the exact algorithm on most problems [69, 78]. This renders HGBC especially appealing to large, high-dimensional data like our balanced N-BaIoT subset.

We measured the average per-sample inference latency and training time of HGBC on our experimental platform to assess the computational practicality of using it to deploy IoT. The hardware used in this research Intel Core i7-7700HQ CPU, 24 GB RAM, a single NVIDIA GTX 1060 (not used in baselines other than software-only baselines), while the software was Python 3.9, scikit-learn/HistGradientBoosting, and NumPy. We report the mean over 5 independent runs; training time is the end-to-end call of fit (excluding loading and pre-processing the dataset). These measurements are meant to give a reproducible measure of the scalability of the model; direct cross-paper comparisons involve the same hardware and measurement methodology, and are thus only reported when the corresponding work actually reports similar measures.

HGBC integrates the benefits of tree ensembles (robustness to feature scaling, natural support of mixed feature types, rapid inference) with computational efficiencies required in large IoT corpora. The feature importance measurements generated by tree-based models also combine well with our SHAP explanations, allowing them to be both highly predictive (test accuracy reported on the prepared N-BaIoT subset) and post-hoc interpretable of the features contributing to the overall classification. The feature ranking based on the Decision-Tree applied in the preprocessing stage (top-20 features) yields features that are especially compatible with tree ensembles. The Histogram Gradient Boosting Classifier is developed for working with both continuous and categorical types of data. The fact that it can accommodate categorical features directly makes it a very suitable tool for a large set of classification tasks. However, every time a machine learning algorithm is adopted, there is a need to have an adequate understanding of the mathematics and principles behind the algorithm in question.

3.4.2 Model Hyperparameters

This subsection lists the Histogram-based Gradient Boosting Classifier (HGBC) hyperparameters used in our experiments, explains their role, and documents the hyperparameter search protocol and final chosen values. All hyperparameters are explicitly reported to allow reproducibility in Table 4.

All these hyperparameters in Table 4 are the key to defining the Hist Gradient Boosting Classifier model's behaviour and are significant to the model. Sensitivity in tuning these hyperparameters will enable a more accurate and refined model to be achieved. The protocol used to optimize the hyperparameters of HGBC was based on stratified 5-

fold cross-validation on the training split alone, with early stopping (validation_fraction = 0.10; max_iter = 100) and randomized search over: learning_rate ∈ (0.05, 0.1, 0.2), max leaf nodes ∈ (15, 31, 63), min samples leaf ∈ (10, 20, 50), and l2 regularization ∈ (0.0, 0.1, 0.5), The final setting reported in section 3.4 was obtained by minimum mean validation log-loss model. The learning rate and max leaf node are used to control the step size; decreasing this value decreases convergence speed and can lead to over-fitting, whereas increasing this value increases convergence speed and may introduce unwanted high variance under class imbalance. L2 regularisation is used to control leaf weights, which prevents spurious partitioning in infrequent cases of traffic, but can lead to unwanted high variance when this term is increased. These decisions, combined with early stopping, reduce overfitting without sacrificing the ability to model multi-class IoT patterns.

Table 4: HGBC hyperparameters, search ranges and final values

Hyperparameter	**Purpose / short description**	**Tuning range (search)**	**Final value**
loss	Loss function for multi-class classification	['categorical_ crossentropy'] (fixed)	categorical_ crossentropy
learning_rate	Shrinkage applied to each tree's contribution (controls step size)	[0.05, 0.1, 0.2]	0.1
max_iter	Maximum number of boosting iterations (trees)	[50, 100, 200]	100
max_leaf_nodes	The maximum number of leaves per tree (controls tree complexity) decision outcome in a decision tree.	[15, 31, 63]	31
max_depth	Maximum depth of individual trees (None = unlimited, limited by max_leaf_nodes)	[None, 6, 10]	None
min_samples_leaf	Minimum number of samples required to form a leaf (regularises splits)	[10, 20, 50]	20
l2_regularization (λ\lambda)	L2 regularisation on leaf weights (stabilises fits)	[0.0, 0.1, 0.5]	0.1
validation_fraction	Fraction of training set used as internal validation for early stopping	[0.05, 0.10]	0.1
random_state	Seed for reproducibility of randomised search and model initialisation	fixed	42

4. RESULT AND DISCUSSION

A Histogram-based Gradient Boosting Classification Tree (HGBC) model has been used. The HGBC is a machine learning model noted for its high throughput and accuracy in categorical and numerical variables. The model of choice was picked for its scalability in dataset size as well as its resistance to overfitting. The method used for implementing involved the data being divided into two sets, these included the training set and the testing set, where the training set was used in feeding the model, while the testing was done on the testing set. It should also be noticed that the performance of the model has been assessed via a number of criteria depending on the problem type, which include accuracy, PR-ROC curve, precision, recall, F1-score, and log loss. This led to the adoption of the HGBC model and, as expected, this produced rather encouraging outcomes.

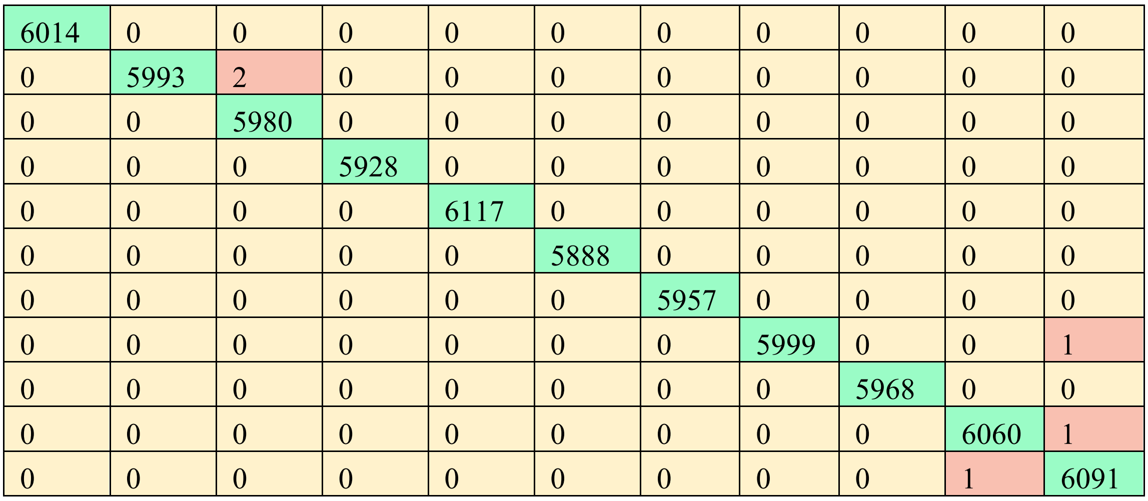
The model thus took roughly 62.88 seconds to run, which is quite reasonable when one considers the nature and intensity of the simulation. The calibration of the HGBC model employed in this study resulted in the highest prediction accuracy of 99 %. Respectively, while the test accuracy was one, indicating a 100% ability of the model to correctly classify almost all its instances in the test set.

We have also tested the use of the Histogram-based Gradient Boosting Classifier (HGBC), which has obtained 100 percent test accuracy in the N-BaIoT dataset. Although this outcome implies good performance, we know that this kind of flawless accuracy may be seen as doubtful. This can be attributed to a number of factors. First, the N-BaIoT dataset includes attack traffic that has widely spaced signatures that are easier to classify compared to mixed or noisy real-world trafficSecond, we employed balanced sampling and SHAP-based feature ranking in order to avoid class and irrelevant feature noise. Thirdly, in order to ensure the model has not merely memorised the training samples, cross-validation alongside early stopping and L2 regularisation were applied in order to fine-tune the hyperparameters. However, we also point out that this is dataset-dependent performance and that the actual IoT traffic can contribute to unknown variations. Hence, we have clearly stated in the limitations section that further analysis of other datasets is required to confirm the generalizability.

In Table 5, we can see that the precision, the recall, as well as the F1-score of the model were 1. This is clear from the classification report, whereby the accuracy achieved for all classes was 1.00 means 100%. The evaluation metrics given below offer a finer view of the effectiveness of the created model, which again proves that the designed model had the capability to classify each class correctly in most of the cases, if not all of them. This is something which is quite commendable as it clearly indicates how effectively the model is able to classify between the different classes. As shown in the confusion matrix in Figure 2, most of the diagonal values are high, indicating that the classification model is doing an excellent job of correctly classifying most of the classes (mainly classes 0-10). However, it is useful to investigate particular study classes with misidentification in more detail. In class 1, two instances were classified as class 2, and in class 2, also two samples were misclassified as class 3. Such misclassifications point to the likelihood of gatekeeper mistakes in differentiating between two neighbouring classes. These kinds of errors might be caused by the similarity or resemblance of characteristics between these classes, and hence call for better feature analysis together with model improvement. We have seen with SHAP analysis that in both Class 1 (BASHLITE scan) and Class 2 (BASHLITE UDP flood), there are some overlapping feature contributions, especially in MI_dir L0.01 meaning and HH L0.1 covariance. This is why this classifier sometimes mixes them up, as scanning traffic can produce burst patterns that look like low-rate UDP floods. In class 10 alone, there is a misclassification whereby one of the instances has been classified as belonging to class 9.

Table 5: The test report of HGBC Classifier

Classes	Precision	Recall	F1-score	Support
0	1.00	1.00	1.00	6014
1	1.00	1.00	1.00	5995
2	1.00	1.00	1.00	5980
3	1.00	1.00	1.00	5928
4	1.00	1.00	1.00	6117
5	1.00	1.00	1.00	5888
6	1.00	1.00	1.00	5957
7	1.00	1.00	1.00	6000
8	1.00	1.00	1.00	5968
9	1.00	1.00	1.00	6061
10	1.00	1.00	1.00	6092
Accuracy	-	-	1.00	66000
Macro avg	1.00	1.00	1.00	66000
Weighted avg	1.00	1.00	1.00	66000



6014	0	0	0	0	0	0	0	0	0	0
0	5993	2	0	0	0	0	0	0	0	0
0	0	5980	0	0	0	0	0	0	0	0
0	0	0	5928	0	0	0	0	0	0	0
0	0	0	0	6117	0	0	0	0	0	0
0	0	0	0	0	5888	0	0	0	0	0
0	0	0	0	0	0	5957	0	0	0	0
0	0	0	0	0	0	0	5999	0	0	1
0	0	0	0	0	0	0	0	5968	0	0
0	0	0	0	0	0	0	0	0	6060	1
0	0	0	0	0	0	0	0	0	1	6091

Figure 2. Confusion Matrix

This difference could be due to some convergence or the classes having some of the species in common, and thus the need to solely clarify and investigate the factors that distinguish them. In this case, further investigation of the SHAP feature based on covariance structures on histogram features demonstrated that the covariance structure between the two classes exists, and the hypothesis of shared traffic signatures is accepted. However, there is a need to undertake a localised evaluation of the types of misclassifications made in interacting with the classes and, more critically, an assessment of the features that may depict the classes' implementation. Modification of the model parameters or exploring other feature selection techniques, or including other data pre-processing techniques, might be necessary to improve the separability of the model. Also, it's important to note that the log loss of the model is 0.0384, which is a performance measure of the current findings on the proposed model. A lower loss means that the model is performing well, and, in this case, the low loss means that the model is making predictions with high confidence.

Table 6 shows the comparison of the various classification techniques employed in a plethora of studies, along with the accuracy obtained while using and not using feature selection, the feature selection method used, whether normalisation was done, whether data was balanced or not and whether a confusion matrix was presented or not. As noted in reference [54] employing the K-Nearest Neighbours (KNN) technique resulted in an accuracy of 99.86% without feature selection. It improves the rates ranging from 6 to 10 percentage points, depending on the adoption of the Mutual Information technique in feature selection and reaches 99.90%. The used data was then made normalised and balanced, and a confusion matrix was also offered. For an 11-class classification task, an Improved Harris Hawks Optimisation technique was employed as cited in the reference [79]. The accuracy level that has been realised through feature selection by means of the FGOA-KNN technique is 98.07%. The data set was pre-processed, including normalisation of data, though no balancing of datasets was done; a confusion matrix was also given. Furthermore, for a 10-class classification in reference [80] the authors implemented the Decision Tree Algorithm that yielded an achieved accuracy of 99.95% accuracy with the two techniques of Data reduction applied through the Features Importance technique. The overall balance of the data was done by using the SMOTE technique, and there was no confusion matrix provided. In reference [81], the model employed was the Random Forest Classifier for the purpose of classification, specifically binary and the given model was found to be precise to 99.98% without feature selection. There was no confusion matrix, which was provided, and the data was normalised and balanced using the SMOTE technique. Also, for the binary classification task, the transformer model, known as The Tab Transformer, introduced in reference [82], was found to have an accuracy rate of 92.33%. Without using a feature selection, the datasets were normalised, although not balanced and no confusion matrix was given. Also, in the work [83], a Convolutional Neural Network (CNN) was applied for binary classification of images with an overall accuracy of 99.87% without feature selection. There were no measures taken to balance the data, nor was the data normalised, and there was no confusion matrix provided. Furthermore, for the 11-class classification task in the referred [84], the Neural Network (NN) has been accomplished with an accuracy of 94.34% without feature selection. No normalisation or balancing of the data was performed, and there was no confusion matrix. In addition, a CNN-LSTM was applied in [85] for the classification of 11 classes, and it got an average of 88.53% accuracy without feature selection. There was no concern for normalisation of data and balancing of the data set, and there was no inclusion of a confusion matrix.

Where the Local-Global Best Bat Algorithm for Neural Networks is used in reference [86] or an 11-class classification task without feature selection, the accuracy achieved was 90%. There was no data normalisation and balancing process; there was no confusion matrix. However, in the work [87] CNN-GRU has been used only for a binary classification problem, and its accuracy was 99.78% were attained with feature selection using the Feature Importance technique. The data was normalised while the data was not balanced; no provision of a confusion matrix was made. Hence, for the binary classification problem in reference [55]. They used a binary classification task resulted in an accuracy of 99.98% The overall accuracy achieved by applying feature selection of the same classifier, although the data was normalised, it was not balanced, and no confusion matrix was given also.

In reference [52] a GXGBoost Model was used for a 3-class classification task, and the accuracy percentage was 99.96%, with feature selection based on the Fisher Score technique. It was confirmed that the data was normalised but not balanced, while a confusion matrix was also given. Next, the XGB-RF was applied in [53] for the classification problem with 11 classes obtained an accuracy of 99.94 % with feature selection using the Recursive Feature Elimination technique. It was quite ambiguous with the data normalisation, but there was no balance, and no confusion matrix was prepared. On the other hand, in reference [88] A DNN-LSTM was used for a 3-class classification task and achieved an accuracy of 99.94% without feature selection. The data was normalised, but it was not balanced, and there was no confusion matrix given. Lastly, the proposed method in this study employed the Histogram Gradient Boosting Classifier for an 11-class classification and obtained an accuracy of 100% with and without using Feature selection using the Technique's Feature Importance. Also, the data was normalised and balanced then a confusion matrix was given.

Table 6: Result evaluation

Ref	Labels	Technique	Acc. no FS.	Acc. + Fs	Fs Technique	Normalize	Balance	Confusion Matrix	Training time (s)	Avg. inference latency (s/sample)
[65]	Binary	KNN	99.86%	99.90%	Mutual Information	Yes	No	Yes	20.622	Not reported
[50]	11-Class	Improved Harris Hawks Optimisation	NA	98.07%	Fgoa-Knn	Yes	No	Yes	65.119	Not reported
[66]	10-class	Decision Tree Algorithm	NA	99.95%	Features Importance	Yes	Smote	No	5.4 – 6.4 s	Not reported
[67]	Binary	Random Forest Classifier	99.98%	NA	NA	Yes	Smote	No	Not reported	Not reported
[68]	Binary	Tab transformer	92.33	NA	NA	Yes	No	No	Not reported	Not reported
[69]	Binary	CNN	99.87%	NA	NA	No	No	No	720 packets/sec	Not reported
[70]	11-Class	NN	94.34%	NA	NA	No	No	No	Not reported	Not reported
[71]	11-Class	Cnn-Lstm	88.53%	NA	NA	No	No	No	Not reported	Not reported
[72]	11-Class	Local-Global Best Bat Algorithm for Neural Networks	90%	NA	Feature Importance	No	No	No	Not reported	Not reported
[73]	Binary	Cnn-Gru	NA	99.78%	Feature Importance	Yes	No	No	8msec |unrealistic	Not reported
[60]	Binary	Logistic Regression	NA	99.98%	ANN	Yes	No	No	Not reported	Not reported
[74]	3-Class	Gxgboost Model	NA	99.96%	Fisher Score	Yes	No	yes	37.496 | 3 features	Not reported
[75]	11-Class	Xgb-Rf	NA	99.94%	Recursive Feature Elimination	Yes	No	No	57.822	0.0010063
[76]	3-Class	Dnn-Lstm	99.94%	NA	NA	Yes	No	No	Not reported	Not reported
Our Mode	11-Class	Histogram Gradient Boosting Classifier	100%	100%	Feature Importance	Yes	Yes	Yes	62.88	0.004

As run-time and latency are significant to the IoT system, we added two additional columns to Table 6, namely Training time (wall-clock seconds) and Avg. inference latency per sample (seconds). In our HGBC experiment, we obtained training time = 62.9 s and an average. Inference latency 8 = 0.004 s/sample on the hardware discussed in section 3.4. In our literature review (Table 6), we discovered that most previous publications report accuracy and preprocessing but do not give end-to-end timing measurements; when the field indicated the computational numbers in the referenced paper, we included them in Table 6; otherwise, the field is marked as Not reported. Since timing varies heavily depending on hardware, the version of software and the details of implementation, we do not perform direct (and possibly inaccurate) numeric comparison unless a paper gives compatible measurements. The figures above demonstrate that HGBC does not merely provide the best predictive performance on N-BaIoT, but also low-latency inference, appropriate for real-time monitoring in highly constrained deployments. It is clear from the set results that the Histogram Gradient Boosting Classifier presented in this research performs the best. This method was able to get a test accuracy of 100% with and without the usage of the Feature Importance technique. This brought about normalisation and balancing of the data, and also a confusion matrix was given. Finally, it can be said that the proposed method has several advantages in comparison with the rest of the methods, in particular, in terms of accuracy. For example, the KNN method applied in refer [54] obtained high accuracy of 99.90% slightly lower than it with a rate of with feature selection. Similarly, the Random Forest Classifier used in reference [81] achieved an accuracy of 99.98% without feature selection, again slightly lower than the proposed method. Even the methods that achieved high accuracies close to 100%, such as the Decision Tree Algorithm in reference [80] with an accuracy of 99.95% with feature selection, and the Logistic Regression in reference [55] with an accuracy of 99.98% with feature selection, did not reach the perfect score achieved by the proposed method. Furthermore, some methods like the Tab Transformer in reference [82] and the CNN-LSTM in reference [85] achieved significantly lower accuracies of 92.33% and 88.53% respectively. Without feature selection, the proposed Histogram Gradient Boosting Classifier demonstrates superior performance in this classification task, achieving the highest accuracy among all the methods compared. This highlights the effectiveness of this method for this specific task. The HGBC model has demonstrated excellent performance on the test data. The low error rate, small leakage, and high accuracy, recall rate, and F1 value confirm that this model is suitable for this classification. Further research might compare how well this model works on other or greater sets of data and also analyse how the efficiency of the model could be increased. This could include changing hyperparameters, using different techniques for feature selection or employing different model structures. All in all, it can be noted that these results are quite promising and indicate that this particular model may be useful for additional classification problems in the future.

5. SHAP (SHAPLEY ADDITIVE EXPLANATIONS) CRITICAL ANALYSIS

In order to get further insights into the performance of the HGBC model and the relevance of the input variables, the authors computed a SHAP analysis. SHAP is a fantastic tool that gives model interpretability based on feature importance in the model's outcome. When using SHAP, the researchers will be able to understand which features are more critical in the HGBC model and thus identify some of the biases present in that model's decision-making process. To go beyond detection accuracy and obtain a more security-relevant insight into how the HGBC model operates and why it made such a decision, we used SHAP (SHapley Additive exPlanations) analysis. SHAP values can be used to obtain a single estimate of feature importance, and also show the extent and direction of each feature contribution to model predictions on individual cases. Figure 3 is a kind of bar chart which is used to illustrate the mean SHAP values for several features in the model based on the different classes. SHAP values are useful in helping to learn about the importance of each feature toward the prediction of the model. The y-axis displays various characteristics of the model, which include: 'MI_dir_L0.1_mean,' MI_dir_L0. 01_variance,' HH_jit_L5_mean', and so on, which may probably be metrics derived from the data/CNs. The x-axis of this plot is as mean SHAP value, which measures the average influence on model output magnitude of the features. Each bar in the bar plot is marked correspondingly to the various classes (Class 0 to Class 10) that exhibit a contribution towards the SHAP value of a specific feature. The feature "MI_dir_L0.1_mean" has the highest mean SHAP value, which shows that the feature has great importance and greatly affects the output of the model, where the influence came mostly from class_9 and class_10. Some of the features, such as "HH_L0.01_covariance","HpHp_L0.01_covariance ", "MI_dir_L0.1_weight" and others, have representations involving more than one class; hence, it is clear that these features are impacted by more than one class. On the other hand, ''HpHp_L5_pcc' and HpHp_L1_pcc display considerably lower mean SHAP, meaning that they are not influential much in the model's prediction.

The SHAP analysis not only prioritises influential features but also provides an insight to understand their practical implications on the security of the IoT. As an example, the feature MI_dir_L0.1_mean, which has the highest SHAP value in the chart, is directional mutual information across flows, meaning that the inbound and outbound flows' traffic patterns rely on each other. Both benign IoT devices and botnets produce fairly consistent dependencies, but since botnets produce highly synchronised or repetitive traffic signals that distort them, they have a relatively stable dependency. Likewise, covariance-based features (e.g., HH_L0.01_covariance, probably based on Hulk attack traffic) represent variation in the shape of the packet-size distributions, which soars dramatically during amplification or flooding attacks. The option HH_jitter_L5_mean illustrates the aspect of jitter (periodic packet timing), which is a well-known feature of volumetric DDoS traffic. SHAP allows security analysts to gain a deeper insight into not only which features are significant but why those features are relevant in identifying what constitutes normal IoT behaviour versus what does not constitute normal behaviour driven by a botnet.

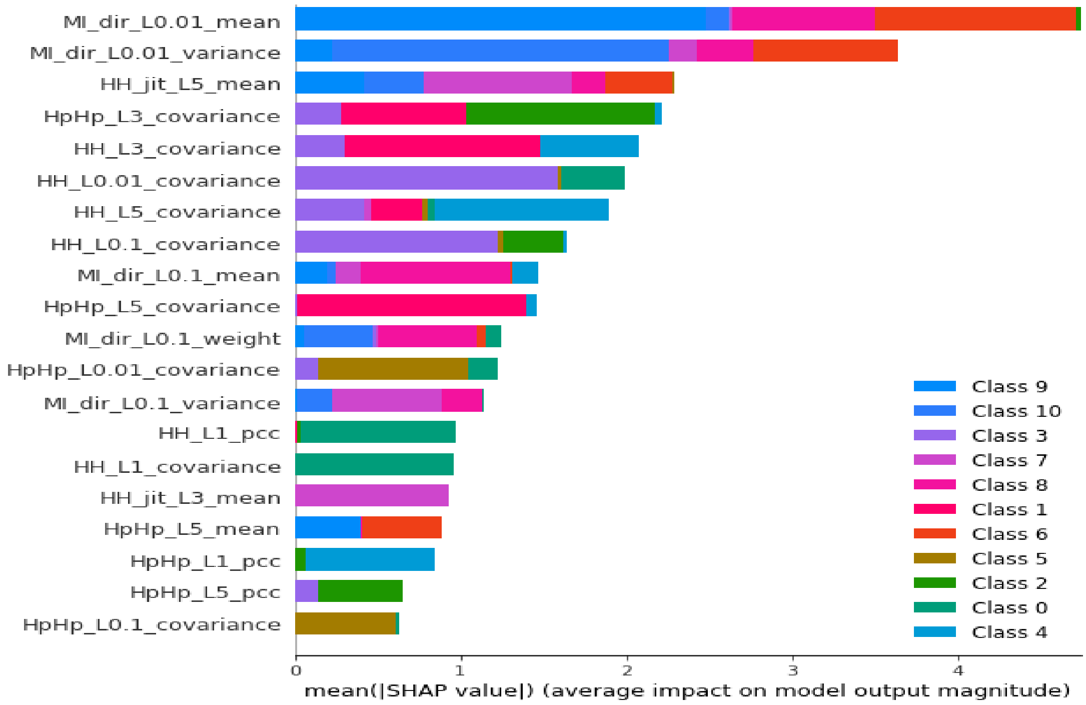


Figure 3. SHAP analysis

Alongside interpretability, SHAP values could be actively taken to improve the detection model. Such features where SHAP contribution is always low (e.g., HpHp_L5_pcc, HpHp_L1_pcc) can be pruned to produce a smaller dimensionality representation without losing accuracy, effectively speeding up the inference performance of the real-time IoT monitoring system. Identical groups of similar high-SHAP features (i.e. the family of directional mutual-information descriptors) could be condensed into simple aggregated indicators to reduce redundancy. Furthermore, SHAP reliance and interdependence plots imply cutoffs and interrelationships between features that may initiate new engineered features to improve segregation between closely similar classes of attacks (e.g., Class 1 vs. Class 2 misclassifications). To reduce the computational cost, we will retrain HGBC on subsets of 15, 30, and 50 top-k SHAP-ranked features in future iterations and check if the reduced-feature models maintain accuracy. This indicates that SHAP not only interpret model results but also optimises and iteratively feature engineers IoT security applications.

The SHAP values for all the features and classes also help in gaining insight into which of the features play an important role and how the different classes impact these features. Such an analysis is useful in selecting and Tuning Features because improved methods can be applied towards enhancing the machine learning model's accuracy while ensuring that it is easy to interpret. Indeed, this chart makes it easy to understand variations and the importance of various features in the model in view of the contribution by different classes. This visualisation is crucial to isolate trends which are critical for understanding the model and its use, as well as for model construction.

6. CONCLUSION AND FUTURE WORK

This paper has shown that the Histogram-based Gradient Boosting Classification Tree (HGBC) model can be used to get very strong results on the N-BaIoT dataset, and get the desired values of perfect precision, recall, F1-score, and nearly zero loss on all classes. The findings validate that HGBC can effectively process categorical and numerical variables, in addition to being superior to most of the reported models in previous studies. Further evidence of the strength of the proposed model was provided by the confusion matrix, indicating that the model is not misclassifying many similar classes. We accept that the results are encouraging, but we do not ignore some significant limitations. First, the N-BaIoT data set is evaluated; there is no confirmation yet that the models are applicable to other IoT threat scenarios. Second, overfitting or biases for this dataset may be present due to the exceptionally high accuracy, which calls for additional verification. Lastly, the reported computational efficiency performance (training and inference time) here is particular to our hardware setup, and should be benchmarked more broadly on shared hardware to compare across the studies. The proposed approach will then be validated by future work on a range of different datasets, including CIC-IoT-2023, Bot-IoT, and TON-IoT, which include more (but not all) types of attacks and traffic conditions. Besides that, we will also grid and Bayesianly optimise HGBC hyperparameters (e.g., max leaf nodes, learning rate, n estimators) to better understand the trade-off between accuracy and execution time. We will also compare other feature selection methods (wrapper-based methods) and the SHAP-driven protocol that is presented in this paper, which consists of the following steps: pruning of low-impact features, aggregation of correlated features, and retraining on top-k SHAP-ranked subsets (k = 15, 30, 50). The purpose of these refinements is to create smaller models with lower latency that can be deployed in real-time Internet of Things security systems.

Moreover, ensemble extensions — for example, stacking HGBC with Random Forest or XGBoost — will be explored to enhance robustness against unseen threats. Lastly, interpretability will continue to be a key component of this work: In order to guarantee that the detection model is not only accurate but also clear and useful for IoT security professionals, SHAP-based explanations will keep providing guidance on feature engineering as well as useful insights into attack behaviour.

REFERENCES

[1] A. Ashraf and W. M. Elmedany, "IoT DDoS attacks detection using machine learning techniques: A Review," in *2021 International Conference on Data Analytics for Business and Industry (ICDABI)*, 2021, pp. 178-185.

[2] S.-H. Lee, Y.-L. Shiue, C.-H. Cheng, Y.-H. Li, and Y.-F. Huang, "Detection and Prevention of DDoS Attacks on the IoT," *Applied Sciences,* vol. 12, p. 12407, 2022.

[3] F. Yousaf, M. Arslan, A. A. Khan, A. Tanzil, A. Batool, and M. Asad, "Machine Learning-Based Detection of Mirai and Bashlite Botnets in IoT Networks," *Journal of Computing & Biomedical Informatics,* vol. 7, pp. 678-689, 2024.

[4] A. Sharma and H. Babbar, "BoT-IoT: Detection of DDoS Attacks in Internet of Things for Smart Cities," in *2023 10th International Conference on Computing for Sustainable Global Development (INDIACom)*, 2023, pp. 438-443.

[5] M. H. Aysa, A. A. Ibrahim, and A. H. Mohammed, "IoT ddos attack detection using machine learning," in *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 2020, pp. 1-7.

[6] S. Mohammed, "A machine learning-based intrusion detection of DDoS attack on IoT devices," *Int. J,* vol. 10, pp. 2278-3091, 2021.

[7] L. S. Vailshery, "Number of IoT connections worldwide 2022-2033, with forecasts to 2030," 2024.

[8] O. Ebrahem, S. Dowaji, and S. Alhammoud, "Towards a minimum universal features set for IoT DDoS attack detection," *Journal of Big Data,* vol. 12, p. 88, 2025.

[9] S. Peddabachigari, A. Abraham, C. Grosan, and J. Thomas, "Modeling intrusion detection system using hybrid intelligent systems," *Journal of network and computer applications,* vol. 30, pp. 114-132, 2007.

[10] A. M. Aleesa and R. Hassan, "A proposed technique to detect DDoS attack on IPv6 web applications," in *2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, 2016, pp. 118-121.

[11] Y. Al-Hadhrami and F. K. Hussain, "DDoS attacks in IoT networks: a comprehensive systematic literature review," *World Wide Web,* vol. 24, pp. 971-1001, 2021.

[12] S. Ikeda, "Iot-based ddos attacks are growing and making use of common vulnerabilities," *URL https://www. cpomagazine. com/cyber-security/iotbased-ddos-attacks-are-growing-and-making-use-of-commonvulnerabilities/(Apr, 2020),* 2020.

[13] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, *et al.*, "N-baiot—network-based detection of iot botnet attacks using deep autoencoders," *IEEE Pervasive Computing,* vol. 17, pp. 12-22, 2018.

[14] A. Petrosyan, "Monthly number of Internet of Things (IoT) malware attacks worldwide from 2020 to 2022," *statista,* Apr 6, 2023 2023.

[15] Z. Chen, F. Jiang, Y. Cheng, X. Gu, W. Liu, and J. Peng, "XGBoost classifier for DDoS attack detection and analysis in SDN-based cloud," in *2018 IEEE international conference on big data and smart computing (bigcomp)*, 2018, pp. 251-256.

[16] A. Alsirhani, S. Sampalli, and P. Bodorik, "Ddos detection system: utilizing gradient boosting algorithm and apache spark," in *2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE)*, 2018, pp. 1-6.

[17] H. Amaad and H. Mughal, "Experimenting Ensemble Machine Learning for DDoS Classification: Timely Detection of DDoS Using Large Scale Dataset," in *2023 4th International Conference on Advancements in Computational Sciences (ICACS)*, 2023, pp. 1-7.

[18] D. Celebucki, M. A. Lin, and S. Graham, "A security evaluation of popular internet of things protocols for manufacturers," in *2018 IEEE International Conference on Consumer Electronics (ICCE)*, 2018, pp. 1-6.

[19] H. Wang, J. Gu, and S. Wang, "An effective intrusion detection framework based on SVM with feature augmentation," *Knowledge-Based Systems,* vol. 136, pp. 130-139, 2017.

[20] M. Saied, S. Guirguis, and M. Madbouly, "A comparative analysis of using ensemble trees for botnet detection and classification in IoT," *Scientific Reports,* vol. 13, p. 21632, 2023.

[21] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran*, et al.*, "Understanding the mirai botnet," in *26th USENIX security symposium (USENIX Security 17)*, 2017, pp. 1093-1110.

[22] K. Angrishi, "Turning internet of things (iot) into internet of vulnerabilities (iov): Iot botnets," *arXiv preprint arXiv:1702.03681,* 2017.

[23] A. B. Mohammed, L. C. Fourati, and A. M. Fakhrudeen, "Isolation Forest Algorithm Against UAV's GPS Spoofing Attack," in *2024 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing & Communications (GreenCom) and IEEE Cyber, Physical & Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics*, 2024, pp. 459-463.

[24] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer,* vol. 50, pp. 80-84, 2017.

[25] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future generation computer systems,* vol. 82, pp. 395-411, 2018.

[26] N. M. Sahar, M. F. S. M. Rozi, N. S. Suriani, S. Sari, S. Ismail, A. A. Jamal*, et al.*, "Advances in DeepFake Detection: Leveraging InceptionResNetV2 for Reliable Video Authentication."

[27] S. Bagui, X. Wang, and S. Bagui, "Machine learning based intrusion detection for IoT botnet," *International Journal of Machine Learning and Computing,* vol. 11, pp. 399-406, 2021.

[28] A. Asokan, "Massive botnet attack used more than 400,000 iot devices," ed: Jul, 2019.

[29] S. A. R. Shah and B. Issac, "Performance comparison of intrusion detection systems and application of machine learning to Snort system," *Future Generation Computer Systems,* vol. 80, pp. 157-170, 2018.

[30] M. Saied and S. Guirguis, "Explainable artificial intelligence for botnet detection in internet of things," *Scientific Reports,* vol. 15, p. 7632, 2025/03/04 2025.

[31] Y. Meidan, M. Bohadana, A. Shabtai, J. D. Guarnizo, M. Ochoa, N. O. Tippenhauer*, et al.*, "ProfilIoT: A machine learning approach for IoT device identification based on network traffic analysis," in *Proceedings of the symposium on applied computing*, 2017, pp. 506-509.

[32] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, D. Breitenbacher, A. Shabtai*, et al.*, "detection_of_IoT_botnet_attacks_N_BaIoT Data Set," *URL: https://archive. ics. uci. edu/ml/datasets/detection_of_IoT_botnet_attacks_N_BaIoT,* 2018.

[33] A. Guryanov, "Histogram-based algorithm for building gradient boosting ensembles of piecewise linear decision trees," in *Analysis of Images, Social Networks and Texts: 8th International Conference, AIST 2019, Kazan, Russia, July 17–19, 2019, Revised Selected Papers 8*, 2019, pp. 39-50.

[34] A. Nazir, J. He, N. Zhu, A. Wajahat, X. Ma, F. Ullah*, et al.*, "Advancing IoT security: A systematic review of machine learning approaches for the detection of IoT botnets," *Journal of King Saud University-Computer and Information Sciences,* vol. 35, p. 101820, 2023.

[35] B. Bala and S. Behal, "AI techniques for IoT-based DDoS attack detection: Taxonomies, comprehensive review and research challenges," *Computer science review,* vol. 52, p. 100631, 2024.

[36] M. A. Hossain, S. Saif, and M. S. Islam, "A novel federated learning approach for IoT botnet intrusion detection using SHAP-based knowledge distillation," *Complex & Intelligent Systems,* vol. 11, pp. 1-23, 2025.

[37] L. C. Guimarães and R. S. Couto, "A performance evaluation of neural networks for botnet detection in the internet of things," *Journal of Network and Systems Management,* vol. 32, p. 98, 2024.

[38] L. L. C. Kasun, Y. Yang, G.-B. Huang, and Z. Zhang, "Dimension reduction with extreme learning machine," *IEEE transactions on Image Processing,* vol. 25, pp. 3906-3918, 2016.

[39] S. Dwivedi, M. Vardhan, and S. Tripathi, "Defense against distributed DoS attack detection by using intelligent evolutionary algorithm," *International Journal of Computers and Applications,* vol. 44, pp. 219-229, 2022.

[40] A. H. Sung and S. Mukkamala, "Identifying important features for intrusion detection using support vector machines and neural networks," in *2003 Symposium on Applications and the Internet, 2003. Proceedings.*, 2003, pp. 209-216.

[41] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on pattern analysis and machine intelligence,* vol. 27, pp. 1226-1238, 2005.

[42] F. Amiri, M. R. Yousefi, C. Lucas, A. Shakery, and N. Yazdani, "Mutual information-based feature selection for intrusion detection systems," *Journal of Network and Computer Applications,* vol. 34, pp. 1184-1199, 2011.

[43] M. Mayuranathan, M. Murugan, and V. Dhanakoti, "Best features based intrusion detection system by RBM model for detecting DDoS in cloud environment," *Journal of Ambient Intelligence and Humanized Computing,* vol. 12, pp. 3609-3619, 2021.

[44] A. Firdaus, N. B. Anuar, A. Karim, and M. F. A. Razak, "Discovering optimal features using static analysis and a genetic search based method for Android malware detection," *Frontiers of Information Technology & Electronic Engineering,* vol. 19, pp. 712-736, 2018.

[45] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in iot," *Sensors,* vol. 17, p. 1967, 2017.

[46] M. Ge, N. F. Syed, X. Fu, Z. Baig, and A. Robles-Kelly, "Towards a deep learning-driven intrusion detection approach for Internet of Things," *Computer Networks,* vol. 186, p. 107784, 2021.

[47] K. Albulayhi, Q. Abu Al-Haija, S. A. Alsuhibany, A. A. Jillepalli, M. Ashrafuzzaman, and F. T. Sheldon, "IoT intrusion detection using machine learning with a novel high performing feature selection method," *Applied Sciences,* vol. 12, p. 5015, 2022.

[48] Z. Ahmad, A. Shahid Khan, K. Nisar, I. Haider, R. Hassan, M. R. Haque*, et al.*, "Anomaly detection using deep neural network for IoT architecture," *Applied Sciences,* vol. 11, p. 7050, 2021.

[49] Q. Abu Al-Haija, "Top-down machine learning-based architecture for cyberattacks identification and classification in iot communication networks," *Frontiers in big Data,* vol. 4, p. 782902, 2022.

[50] Q. Abu Al-Haija and A. Al-Badawi, "Attack-Aware IoT network traffic routing leveraging ensemble learning," *Sensors,* vol. 22, p. 241, 2021.

[51] C. D. McDermott, F. Majdani, and A. V. Petrovski, "Botnet detection in the internet of things using deep learning approaches," in *2018 international joint conference on neural networks (IJCNN)*, 2018, pp. 1-8.

[52] M. Alqahtani, H. Mathkour, and M. M. Ben Ismail, "IoT botnet attack detection based on optimized extreme gradient boosting and feature selection," *Sensors,* vol. 20, p. 6336, 2020.

[53] J. A. Faysal, S. T. Mostafa, J. S. Tamanna, K. M. Mumenin, M. M. Arifin, M. A. Awal*, et al.*, "XGB-RF: A hybrid machine learning approach for IoT intrusion detection," in *Telecom*, 2022, pp. 52-69.

[54] M. Al-Sarem, F. Saeed, E. H. Alkhammash, and N. S. Alghamdi, "An aggregated mutual information based feature selection with machine learning methods for enhancing IoT botnet attack detection," *Sensors,* vol. 22, p. 185, 2021.

[55] F. Abbasi, M. Naderan, and S. E. Alavi, "Anomaly detection in Internet of Things using feature selection and classification based on Logistic Regression and Artificial Neural Network on N-BaIoT dataset," in *2021 5th International Conference on Internet of Things and Applications (IoT)*, 2021, pp. 1-7.

[56] H. Wasswa, H. Abbass, and T. Lynar, "Are GNNs Worth the Effort for IoT Botnet Detection? A Comparative Study of VAE-GNN vs. ViT-MLP and VAE-MLP Approaches," *arXiv preprint arXiv:2505.17363,* 2025.

[57] A. Naeem, M. A. Khan, N. Alasbali, J. Ahmad, A. A. Khattak, and M. S. Khan, "Efficient IoT Intrusion Detection with an Improved Attention-Based CNN-BiLSTM Architecture," *arXiv preprint arXiv:2503.19339,* 2025.

[58] R. Kalakoti, H. Bahsi, and S. Nõmm, "Explainable federated learning for botnet detection in iot networks," in *2024 IEEE International Conference on Cyber Security and Resilience (CSR)*, 2024, pp. 01-08.

[59] P. K. Myakala, S. Kamatala, and C. Bura, "Privacy-Preserving Federated Learning for IoT Botnet Detection: A Federated Averaging Approach," 2025.

[60] K. A. Alaghbari, H.-S. Lim, M. H. M. Saad, and Y. S. Yong, "Deep autoencoder-based integrated model for anomaly detection and efficient feature extraction in iot networks," *IoT,* vol. 4, pp. 345-365, 2023.

[61] T. A. Tuan, H. V. Long, L. H. Son, R. Kumar, I. Priyadarshini, and N. T. K. Son, "Performance evaluation of Botnet DDoS attack detection using machine learning," *Evolutionary Intelligence,* vol. 13, pp. 283-294, 2020.

[62] D. Acarali and M. Rajarajan, "Botnet-based attacks and defence mechanisms," *Versatile Cybersecurity,* pp. 169-199, 2018.

[63] D. Dua and C. Graff, "UCI machine learning repository," 2017.

[64] A. Marzano, D. Alexander, O. Fonseca, E. Fazzion, C. Hoepers, K. Steding-Jessen*, et al.*, "The evolution of bashlite and mirai iot botnets," in *2018 IEEE Symposium on Computers and Communications (ISCC)*, 2018, pp. 00813-00818.

[65] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert systems with applications,* vol. 73, pp. 220-239, 2017.

[66] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of big data,* vol. 6, pp. 1-54, 2019.

[67] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset," *Future Generation Computer Systems,* vol. 100, pp. 779-796, 2019.

[68] R. L. Figueroa, Q. Zeng-Treitler, S. Kandula, and L. H. Ngo, "Predicting sample size required for classification performance," *BMC medical informatics and decision making,* vol. 12, p. 8, 2012.

[69] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785-794.

[70] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-validation," in *Encyclopedia of database systems*, ed: Springer, 2009, pp. 532-538.

[71] G. Louppe, *Understanding random forests: From theory to practice*: Universite de Liege (Belgium), 2014.

[72] P. M. Granitto, C. Furlanello, F. Biasioli, and F. Gasperi, "Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products," *Chemometrics and intelligent laboratory systems,* vol. 83, pp. 83-90, 2006.

[73] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research,* vol. 3, pp. 1157-1182, 2003.

[74] S. Narla, S. Peddi, and D. T. Valivarthi, "Optimizing predictive healthcare modelling in a cloud computing environment using histogram-based gradient boosting, MARS, and SoftMax regression," *International Journal of Management Research and Business Strategy,* vol. 11, pp. 25-40, 2021.

[75] N. L. Fitriyani, M. Syafrudin, N. Chamidah, M. Rifada, H. Susilo, D. Aydin*, et al.*, "A Novel Approach Utilizing Bagging, Histogram Gradient Boosting, and Advanced Feature Selection for Predicting the Onset of Cardiovascular Diseases," *Mathematics,* vol. 13, p. 2194, 2025.

[76] P. Theerthagiri, "Liver disease classification using histogram-based gradient boosting classification tree with feature selection algorithm," *Biomedical Signal Processing and Control,* vol. 100, p. 107102, 2025.

[77] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics,* pp. 1189-1232, 2001.

[78] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma*, et al.*, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems,* vol. 30, 2017.

[79] F. Taher, M. Abdel-salam, M. Elhoseny, and I. M. El-hasnony, "Reliable Machine Learning Model for IIoT Botnet Detection," *IEEE Access,* 2023.

[80] S. Popoola, R. Ande, A. Atayero, M. Hammoudeh, G. Gui, and B. Adebisi, "Optimized Lightweight Federated Learning for Botnet Detection in Smart Critical Infrastructure," 2023.

[81] M. G. Karthik and M. M. Krishnan, "Hybrid random forest and synthetic minority over sampling technique for detecting internet of things attacks," *Journal of Ambient Intelligence and Humanized Computing,* pp. 1-11, 2021.

[82] A. KALIDINDI and M. B. ARRAMA, "A TABTRANSFORMER BASED MODEL FOR DETECTING BOTNET-ATTACKS ON INTERNET OF THINGS USING DEEP LEARNING," *Journal of Theoretical and Applied Information Technology,* vol. 101, 2023.

[83] M. Gromov, D. Arnold, and J. Saniie, "Edge Computing for Real Time Botnet Propagation Detection," in *2022 IEEE International Conference and Expo on Real Time Communications at IIT (RTC)*, 2022, pp. 13-16.

[84] S. Kalenowski, D. Arnold, M. Gromov, and J. Saniie, "Heterogeneity Tolerance in IoT Botnet Attack Classification," in *2023 IEEE International Conference on Electro Information Technology (eIT)*, 2023, pp. 353-356.

[85] H. Alkahtani and T. H. Aldhyani, "Botnet attack detection by using CNN-LSTM model for Internet of Things applications," *Security and Communication Networks,* vol. 2021, pp. 1-23, 2021.

[86] A. Alharbi, W. Alosaimi, H. Alyami, H. T. Rauf, and R. Damaševičius, "Botnet attack detection using local global best bat algorithm for industrial internet of things," *Electronics,* vol. 10, p. 1341, 2021.

[87] Z. Wang, H. Huang, R. Du, X. Li, and G. Yuan, "IoT Intrusion Detection Model based on CNN-GRU," *Frontiers in Computing and Intelligent Systems,* vol. 4, pp. 90-95, 2023.

[88] T. Hasan, J. Malik, I. Bibi, W. U. Khan, F. N. Al-Wesabi, K. Dev*, et al.*, "Securing industrial internet of things against botnet attacks using hybrid deep learning approach," *IEEE Transactions on Network Science and Engineering,* 2022.