International Journal of Innovative Research and Scientific Studies, 8(2) 2025, pages: 338-347

IJIRSS
International Journal of Innovative Research & Scientific Studies

ISSN: 2617-6548

URL: www.ijirss.com

Evolving botnet defenses: A survey of machine learning approaches for identifying polymorphic and evasive malware

Sina Ahmadi

National Coalition of Independent Scholars, USA.

(*Email: sina0@acm.org*)

Abstract

The advancement of polymorphic and evasive malware helps botnets overcome traditional security mechanisms, rendering them obsolete. This fact, along with the sophisticated growth of botnets, poses a threat to modern computer networks. As cyber threats evolve, so must the strategies used to detect and mitigate them. This paper highlights the various machine learning (ML) techniques employed for botnet detection, outlining their advantages, limitations, and practical applications. The study analyzes supervised, unsupervised, and deep learning approaches and examines their role in detecting malicious network behavior. It is discovered that although the ML-based detection systems provide promising solutions, exposing the detection system to a real-world scenario uncovers more issues like adversarial resistance, scalability, and computational overhead. Furthermore, this paper brings attention to new issues such as providing strong defenses against adversarial attacks and the use of explainable AI for a better understanding of their purpose. With the goal of improving the state of botnet defense, this research aims to provide comprehensive methodologies while underscoring existing gaps toward ensuring continuous development in robust cybersecurity strategies driven by machine learning.

Keywords: Botnets, Machine learning, Cybersecurity, DDoS, Deep learning, Privacy, XAI, ZTA.

DOI: 10.53894/ijirss.v8i2.5163
Funding: This study received no specific financial support.
History: Received: 10 January 2025 / **Revised:** 12 February 2025 / **Accepted:** 17 February 2025 / **Published:** 7 March 2025

Competing Interests: The author declares that there are no conflicts of interests regarding the publication of this paper.
Transparency: The author confirms that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.
Publisher: Innovative Research Publishing

1. Introduction

The rapid evolution and increasing sophistication of botnets have made them one of the most pressing threats to computer systems and networks. Botnets are networks of compromised computers or devices, collectively controlled by an attacker to execute malicious activities such as distributed denial-of-service (DDoS) attacks, phishing campaigns, spamming, and credential theft [1-3]. The widespread adoption of Internet of Things (IoT) devices has further exacerbated the problem, providing attackers with an extensive attack surface [4, 5].

Polymorphic and evasive malware, integral components of modern botnets, add another layer of complexity by altering their behavior and appearance to evade detection [2, 6, 7]. Traditional security measures, such as signature-based detection

systems, often fail against such sophisticated threats, as these systems rely on predefined patterns or signatures that can be easily circumvented by adaptive malware [8, 9]. This inadequacy has prompted the cybersecurity community to explore more advanced detection mechanisms.

Distributed identity represents a major shift in digital identity management, providing a decentralized model that aligns with ZTA principles. Unlike conventional systems, which rely on centralized data centers, distributed identity allows individuals to own and manage their identity information [3]. Moreover, the detection of unusual network behavior is often done using clustering algorithms through unsupervised learning techniques [10]. The invention of deep learning convolutional and recurrent neural networks has opened new avenues for capturing sophisticated temporal and spatial structures in network traffic which further enhances botnet detection [6, 7].

Botnets have significantly evolved since their initial use for DDoS attacks in the early 2000s [11]. Over time, they have adopted advanced techniques such as peer-to-peer (P2P) architectures, domain generation algorithms (DGAs), and fast-flux DNS techniques, enabling them to maintain resilience and evade takedowns [12, 13]. These advancements have made it more challenging to detect and eliminate botnets [2, 3].

ML-based detection offers plenty of opportunities, but there are still shortcomings. Models require sufficient quantities of varied training datasets, and acquiring them is challenging [4]. In addition, ML models can be compromised with adversarial attacks, where the data presented to the model is sabotaged in order to avoid detection [2, 6]. Issues of overfitting and massive computational overhead are also very problematic [7].

This survey aims to analyze the most advanced ML approaches for botnet detection in detail, with an emphasis on what they can or cannot achieve and what needs to be done to meet the changing challenges. Through the study of past and recent developments, this paper attempts to identify the gaps that need attention to foster innovation for botnet defense.

2. Literature Review

For over two decades, botnets have emerged as a serious danger to various computer systems and networks [8, 9]. Their progression has been marked by advancements in architecture, communication, and evasion techniques. The earlier forms of botnets used centralized command and control (C&C) servers and Internet Relay Chat (IRC) as the medium of communication [11, 12]. These systems were, and remain, remarkably inefficient because they were simple to detect and dismantle, thanks to the centralized structure that depended on a single point of failure.

Modern botnets, however, have adopted advanced mechanisms that make detection and mitigation far more challenging. Some of the most prominent techniques include:

1. Peer-to-Peer (P2P) Architectures: P2P architectures decentralize control, distributing it across the network. This eliminates the single point of failure inherent in centralized systems, making it significantly harder to disable the botnet by targeting individual bots or nodes [13].
2. Domain Generation Algorithms (DGAs): DGAs dynamically generate new domain names for command and control (C&C) communication, allowing botnets to bypass domain blacklisting. This ensures that the botnet remains operational even if some domains are blocked [12].
3. Fast-Flux DNS Techniques: By rapidly changing the IP addresses associated with a domain name, fast-flux techniques make it difficult for security systems to track or block C&C servers. These techniques often employ compromised hosts as proxies to further obscure communication channels.
4. Sophisticated Encryption and Tunneling Mechanisms: Modern botnets frequently use strong encryption protocols and tunneling mechanisms to disguise their traffic. This hinders the ability of network security tools to inspect and analyze malicious communications.
5. Advanced Polymorphic and Metamorphic Capabilities: Polymorphic malware modifies its code structure with each iteration, while metamorphic malware rewrites its own code entirely. These techniques allow botnets to evade signature-based detection by constantly altering their appearance.

Such advancements allow botnets to adapt their behavior and avoid being detected [2, 3]. Modern botnets can bypass standard security defenses through the use of a number of obfuscation techniques, such as code obfuscation or anti-debugging, along with sandbox evasion and other advanced techniques [2, 6, 7].

To tackle and remedy these issues, many researchers have increasingly turned to machine learning (ML)-based approaches to improve botnet detection accuracy. For instance, Binkley and Singh [14] proposed a novel approach for detection that is based on artificial neural networks trained on network activity data. Gu, et al. [15] created BotHunter, which is an AI-powered framework designed for the detection of botnet-associated traffic. Strayer, et al. [16] suggested analyzing the traffic pattern in terms of its flow and using AI to identify the presence of botnets in the traffic.

Recent studies build on the methodologies of their predecessors, broadening the possibilities of AI in detecting botnets. For example, Kumar, et al. [1] twined the classification of botnet traffic within these patterns with supervised learning techniques, especially with random forests. Further, Li, et al. [2] confirmed the possibility of using artificial convolution neural networks (CNN) for botnet detection while Santos, et al. [3] utilized manual techniques like clustering k-means for irregular traffic pattern detection. All of these studies highlight the versatility and effectiveness of applying machine learning to botnet attack detection and takedown.

3. Machine Learning Approaches for Botnet Detection

ML approaches have become a crucial part of research in botnet detection due to their capability to process a large volume of data and discover details that other approaches tend to overlook [2, 6, 7]. ML techniques are divided into three

broad categories: supervised learning, unsupervised learning, and deep learning methods, all of which have specific merits in botnet activity monitoring.

3.1. Supervised Learning Methods

Supervised learning trains an ML algorithm on labeled datasets, wherein each sample is associated with a known target label. This allows the algorithm to understand the relationship between input features and output labels so that it can classify new, unseen data appropriately [1, 2].

In botnet detection, supervised learning methods such as Support Vector Machines (SVMs), Random Forests, and Gradient Boosting Machines (GBMs) are popular and have been widely used. SVMs work extremely well on high-dimensional data, they are particularly effective in separating normal and malicious traffic [6]. On the other hand, Random Forests have shown to be highly accurate for botnet tasks and are significantly less prone to overfitting [7]. As an example, Li, et al. [2] report the use of GBM, which in its combined approach also known as ensemble learning, combines many weak classifiers for improved optimal detection rates. Kumar, et al. [1] are one of the many who applied Random Forests on network traffic and showed competitive accuracy in botnet detection.

3.2. Unsupervised Learning Methods

In contrast to supervised techniques, unsupervised learning does not make use of labeled datasets. Rather, it seeks to find patterns, outliers, or clusters within the datasets [1, 2]. This characteristic gives it an edge in situations where datasets with labels do not exist.

K-means Clustering, Hierarchical Clustering, and Principal Component Analysis (PCA) are three broad categories of unsupervised techniques used for botnet detection. K-means clustering is popular for classifying similar traffic patterns that help in detecting certain botnet-related malicious behaviors [6]. In their work, Santos, et al. [3] revealed the potential of k-means clustering in detecting botnet traffic with high accuracy rates. Hierarchical Clustering which builds a tree-like structure of clusters is useful in revealing multi-layered botnet traffic patterns [7]. PCA, by reducing dimensionality, increases the chances of detecting subtle anomalies in massive amounts of network data [2].

3.3. Deep Learning Methods

Deep Learning is an advanced subset of ML algorithms that employ neural networks to analyze complex relationships in data [2, 6, 7]. With the help of big data, deep learning algorithms manage to capture spatial and temporal patterns in network traffic, thus aiding in effective botnet detection.

Popular algorithms that are used include Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM). CNNs focus on detecting spatial features embedded within network traffic data whereas RNNs and LSTMs focus on temporal dependencies, which are sequentially organized pieces of data revolving around certain activities within a network. With this more in-depth approach, deep learning models are able to understand bots more comprehensively, which makes them crucial for detecting polymorphic and evasive malware in networks.

3.4. Trade-offs of Machine Learning Approaches

Machine Learning approaches for botnet detection have their advantages and limitations based on the adopted algorithm.

- Supervised Learning: The effectiveness of supervised learning techniques in botnet detection stems from their accuracy in recognizing unique patterns in labeled datasets. Practitioners and researchers often prefer using algorithms such as Support Vector Machines (SVMs) and Random Forests due to their ease of use and understanding [1, 2]. On the other hand, the reliance on labeled data is a drawback in and of itself, since gathering quality labeled datasets, particularly concerning botnet traffic, is incredibly laborious and costly [7]. Moreover, these models may not perform adequately in terms of generalization when presented with new or heavily skewed datasets. The use of Adversarial Attacks, in which attackers covertly change input data so that they remain
- undetected, only serves to expose the weaknesses the supervised approaches face [6].
- Unsupervised Learning: In situations where data sets lack labels, unsupervised learning approaches tend to work better. k–means clustering and Principal Component Analysis (PCA) are some of the techniques that have been very effective in detecting intrusions, particularly anomalies and outliers in network traffic [3, 6]. They are versatile in that they can incorporate new patterns that were not previously available, thus enabling them to detect new botnet behaviors. Their downside is that they do not work very well unless the feature selection and the preprocessing stage are done well. Furthermore, unsupervised learning techniques are often less accurate than supervised techniques, less accepted because the results are harder to understand, and less efficient in practical applications [7].
- Deep Learning: Deep learning techniques serve as the state of the art in ML-based botnet detection given their prowess in modeling complex temporal and spatial interactions in network data [1, 2]. Some algorithms like Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks can identify polymorphic as well as evasive stealthy malware with high accuracy. The major puzzle is that deep learning relies heavily on the availability of large-scale datasets, which makes its application challenging. Moreover, acquiring the hardware needed to train and implement these models is expensive. These models are also prone to overfitting, especially when there is a lack of variety in the training data, which is compounded by the fact that their use brings a lot of complexities which inherently makes them less interpretable [2].

Key strengths and weaknesses of supervised, unsupervised, and deep learning approaches are summarized in Table 1.

Table 1.
Strengths and Weaknesses of ML Approaches.

ML Approach	Strengths	Weaknesses
Supervised Learning	- High accuracy for labeled datasets - Easy to implement and interpret Well-suited for specific classification tasks	- Requires a large, labeled dataset - Limited generalization to unseen or imbalanced data Vulnerable to adversarial attacks
Unsupervised Learning	- Does not rely on labeled data - Effective for anomaly and outlier detection - Flexible in identifying previously unseen patterns	- Lower accuracy compared to super- vised methods - Sensitive to feature selection and pre-processing Results can be hard to interpret
Deep Learning	- High accuracy in detecting complex patterns - Capable of modeling temporal and spatial relationships Automatically extracts features	- Requires large datasets and significant computational resources - Prone to overfitting without careful tuning Difficult to interpret and explain the results

4. Methods and Results Comparison

Several research studies have evaluated different ML algorithms for botnet detection using the CTU-13, ISCX, and Bot-IoT datasets. These studies report varying accuracies depending on the specific algorithm and dataset used [4, 17, 18].

4.1. CTU-13 Dataset

- Decision Tree: Garcia, et al. [19] achieved 98.7% accuracy while Shiravi, et al. [18] reported an accuracy of 96.91% using DTs on this dataset.
- Naive Bayes: Garcia, et al. [19] reported an accuracy of 96.25%, while Koroniotis, et al. [4] reported 98.5%.
- K-Nearest Neighbors: Garcia, et al. [19] achieved 90.80% accuracy, and Tongun [20] reported 96.24%.
- Support Vector Machine: Shiravi, et al. [18] achieved an accuracy of 96.43%, while Koroniotis, et al. [21] reported 99.5%, and Moustafa, et al. [5] reported 98.80%.
- Neural Networks: Moustafa, et al. [5] reported an accuracy of 99.97%. Mirza [22] demonstrated a 99.2% prediction accuracy.
- M-Means Clustering: Another study by University [23] demonstrated a detection accuracy rate of 97.11%.
- Random Forest: Koroniotis, et al. [21] achieved an accuracy of 96.41% using Random Forest.

4.2. ISCX-12 Dataset

- Decision Tree: Shiravi, et al. [18] achieved an accuracy of 95.3% using DTs on this dataset.
- Naive Bayes: Shiravi, et al. [18] reported an accuracy of 92.3% for detecting malware traffic using NB on this dataset.
- K-Nearest Neighbors: Bakker [10] reported KNN to yield an accuracy of 94.4%.
- Support Vector Machine: Moustafa, et al. [5] achieved 91% accuracy, while Sharafaldin, et al. [24] reported 93.7%.
- Random Forest: Shiravi, et al. [18] reported 96% accuracy, while Koroniotis, et al. [4] achieved nearly 98%.
- Extra Trees: Shiravi, et al. [18] achieved 96% accuracy, and Bakker [10] reported 97.5%.
- XGBoost: Shiravi, et al. [18] achieved an accuracy of 95.2%, while Moustafa, et al. [5] reported 95%.
- Graph Neural Networks: Koroniotis, et al. [4] achieved 98.36% accuracy in identifying malicious traffic.
- K-Means Clustering: A study by the University [23] achieved a detection accuracy rate of 90.68%, and Bakker [10] reported 97.11%.

4.3. Bot-IoT Dataset

- Random Forest: Moustafa, et al. [5] achieved 97% accuracy, while Koroniotis, et al. [25] reported 99.9978%.
- Logistic Regression: Atuhurra, et al. [26] reported a 99.63% accuracy.
- Deep Neural Networks: A study by the UNSW University Cyber [27] achieved accuracies close to 99.8%.
- K-Means Clustering: A Purdue University's study University [23] achieved a detection rate of 97.11%.
- Hierarchical Clustering: Yin, et al. [28] reported high detection accuracy rate of 99.89% through Hierarchical Clustering on this dataset.

These studies often utilize various performance metrics besides accuracy, including precision, recall, F1-score, and AUC-ROC. For simplicity, we only look at accuracy explicitly. The choice of features and hyperparameter optimization also play significant roles in the reported performance [4, 5].

Each of these datasets has its own strengths and weaknesses, and the choice of dataset depends on the specific requirements of the detection system being evaluated. A comparison of the different datasets used for botnet detection is shown in Table 2.

Table 2.
Comparison of machine learning approaches for botnet detection across datasets.

Algorithm	CTU-13 Dataset (%)	ISCX-2012 Dataset (%)	Bot-IoT Dataset (%)	References
Decision Tree	98.7, 96.91	95.3	-	Shiravi, et al. [18] and Garcia, et al. [19]
Naive Bayes	96.25, 98.5	92.3	-	Koroniotis, et al. [4]; Shiravi, et al. [18] and Garcia, et al. [19]
K-Nearest Neighbors	90.8, 96.24	94.4	-	Bakker [10]; Garcia, et al. [19] and Tongun [20]
Support Vector Machine	96.43, 99.5, 98.8	91, 93.7	-	Moustafa, et al. [5]; Shiravi, et al. [18]; Koroniotis, et al. [21] and Sharafaldin, et al. [24]
Neural Networks	99.97, 99.2	-	99.8	Moustafa, et al. [5]; Mirza [22] and Cyber [27]
Random Forest	96.41	96, 98	97, 99.9978	Koroniotis, et al. [4]; Moustafa, et al. [5]; Shiravi, et al. [18] and Koroniotis, et al. [21]
Extra Trees	-	96, 97.5	-	Bakker [10] and Shiravi, et al. [18]
XGBoost	-	95.2, 95	-	Moustafa, et al. [5] and Shiravi, et al. [18]
K-Means Clustering	97.11	90.68, 97.11	97.11	Bakker [10] and University [23]
Hierarchical Clustering	-	-	99.89	Yin, et al. [28]
Logistic Regression	-	-	99.63	Atuhurra, et al. [26]

From the results presented, it is clear that different machine learning algorithms excel depending on the dataset. For the CTU-13 dataset, Neural Networks achieved the highest accuracy of 99.97%, demonstrating their ability to model complex network traffic patterns effectively. For the ISCX-12 dataset, Random Forest emerged as a leading approach, achieving an accuracy of 98% due to its robustness and capacity to handle imbalanced data effectively. On the Bot-IoT dataset, Logistic Regression displayed remarkable performance, achieving 99.63%, showcasing that even simpler algorithms can yield high accuracy when applied to structured and well-prepared datasets.

Even as Neural Networks offer unrivaled accuracy for classification tasks on the CTU-13 dataset, the dependence on computational power and large datasets remains a gap. Random Forests and Logistic Regression are less complex, yet provide competitive accuracy on the ISCX-12 and Bot-IoT datasets, demonstrating their strength and flexibility. Furthermore, the effectiveness of Logistic Regression on the Bot-IoT dataset points to the importance of simpler available algorithms when resources are limited. All these results show that the concept of the 'best' algorithm is purely relative, influenced by the nature of the dataset and real-life constraints.

6. State-of-the-Art Analysis

Recent developments in machine learning-based botnet detection continue to demonstrate substantial progress in both detection accuracy and efficiency. Table 3 summarizes the state-of-the-art approaches for botnet detection, highlighting their architectures, datasets, accuracies, and associated references.

The state-of-the-art analysis reveals key advancements:

- CTU-13 Dataset: Methods such as the Self-Organizing Map (SOM) and CatBoost Classifier have achieved exceptional accuracy, surpassing 99.8%, while DeepDefense demonstrated strong performance with 98.3% accuracy.
- ISCX-2012 Dataset: Hybrid approaches combining clustering and classification (e.g., K-Means with Naive Bayes) and ensemble methods like BotHunter+ have achieved accuracies of 99% and 96.8%, respectively.
- Bot-IoT Dataset: Advanced architectures such as SMOTE-DRNN, IDBO-CatBoost, and ZOA+DGAN have consistently exceeded 98.5%, with ZOA+DGAN achieving the highest accuracy of 99.87%.

This analysis highlights that across all datasets, advanced techniques like CatBoostClassifier and ZOA+DGAN stand out for their exceptional accuracy, achieving 99.87% on the CTU-13 and Bot-IoT datasets, respectively. CatBoost excels in handling categorical features efficiently, making it suitable for diverse network traffic data, while ZOA+DGAN leverages feature selection and adversarial training to address evolving botnet behaviors. The hybrid K-Means and Naive Bayes approach, achieving 99.0% accuracy on the ISCX-12 dataset, illustrates the potential of combining clustering and classification methods. These results suggest that while dataset-specific optimizations are critical, integrating feature engineering, boosting techniques, and hybrid architectures may provide a universal edge for botnet detection.

Table 3.
State-of-the-Art Analysis for Botnet Detection.

Method	Architecture	Dataset	Accuracy (%)	Reference
DeepDefense	CNN+LSTM	CTU-13	98.3	Chen, et al. [29]
Self-Organizing Map (SOM)	SOM-based traffic analysis	CTU-13	99.78	Kohonen and Lehtokangas [30]
CatBoostClassifier	Gradient Boosting Decision Tree	CTU-13	99.87	Prokhorenkova, et al. [31]
BotHunter+	Random Forest + XGBoost	ISCX-2012	96.8	Alissa, et al. [32]
K-Means + Naive Bayes	Hybrid clustering-classification	ISCX-2012	99.0	Beigi, et al. [33]
SMOTE-DRNN	SMOTE + Deep Residual Neural Network	Bot-IoT	99.75	Wang, et al. [34]
IDBO-CatBoost	CatBoost with Bayesian Optimization	Bot-IoT	98.57	Chen and Li [35]
Voting Ensemble	Ensemble Voting Classifier	Bot-IoT	99.0	Patel and Desai [36]
ZOA+DGAN	Feature Selection + Deep Generative Adversarial Network	Bot-IoT	99.87	Singh, et al. [37]

Although these techniques accomplish significant accuracy, issues like operational costs, dependence on extensive datasets, and flexibility to novel attacks remain. These issues, however, are greatly exacerbated in resource-limited settings such as IoT networks, where performance and cost-effectiveness are strongly constrained. To fill these gaps, there is a need to design lightweight, efficient models that allow for immediate use. Moreover, witnessing the advancement of techniques that botnets use, especially adversarial approaches that attempt to evade detection systems, has shown the need for greater proactive and adaptive systems. There has been a shift in the focus of researchers towards hybrid systems, more sophisticated feature engineering, and explainable models to tackle these problems effectively and in a manner that is geared toward system scalability. This last claim is further substantiated by the following section, which discusses emerging areas of interest capturing the tremendous effort directed towards innovations in ML-based botnet detection.

7. Emerging Areas of Interest

Due to the pressing need to deal with the evolving mechanisms of botnets, machine learning-driven botnet detection is progressing rapidly. The focus on increasing accuracy and efficiency and countering complex threats are some of the prominent research areas that are affecting the progress of the field.

7.1. Converged Systems

Using multiple machine-learning approaches together can improve botnet detection. Hybrid systems combine deep learning, like CNNs, with traditional methods such as Random Forests or SVMs for classification. These systems are reported to boost accuracy by 15%–20%. For example, CNNs can extract detailed spatial and temporal features from botnet traffic, which Random Forests then classify [1, 29]. Combining deep learning with traditional methods improves feature detection while making models more interpretable.

7.2. Real-Time Detection

Real-time detection is becoming increasingly important due to the growing frequency and severity of bot attacks. Stream processing and online learning have reduced the time needed to process new data by 40%–60%. These systems constantly adapt, remaining effective against evolving threats. Additionally, lightweight, energy-efficient architectures are crucial for use on IoT devices and other resource-limited environments.

7.3. Advanced Feature Engineering

Machine learning models for detecting botnets rely heavily on effective feature engineering. Recent advances in automated feature selection and extraction have improved system efficiency by 25%–30% [34]. Techniques like mutual information-based feature selection and domain-specific feature specialization help process large amounts of network traffic and identify key features. For example, using deep feature representation learning and mutual information reduces the need for manual feature engineering, speeding up model development.

7.4. Explainability and Interpretability

Machine learning in security-critical applications faces challenges due to its lack of transparency. Efforts are growing to make ML models for botnet detection more explainable. Tools like SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations) help build trust by clarifying how models make predictions [36]. Explainable AI (XAI) is essential not only for debugging but also for ensuring that models comply with cybersecurity regulations.

7.5. Adversarial Robustness

Combining botnet detection with adversarial machine learning has become crucial. Attackers now create sophisticated techniques designed to bypass ML-based detection systems. In response, researchers are developing Adversarially Defendable Robust Models (ADRM). These models use adversarial examples during training and strengthen the model with special techniques, such as robust loss functions, to improve defenses [37]. For machine learning models to be reliable in real-world applications, they must be able to resist such adversarial attacks.

8. Challenges

8.1. Data Challenges

- Limited availability of labeled data can hinder effective training and evaluation of ML models.
- Poor data quality can lead to subpar model performance and inaccurate results.
- The collection and use of data raise significant privacy concerns, especially in environments with sensitive data.
- Ensuring the security of data used in ML-based botnet detection systems is critical, particularly in environments with high-value data.

8.2. Cost Challenges

The cost of implementing and maintaining ML-based botnet detection systems can be significant. According to a Gartner report, Gartner [38] the average cost of implementing an ML-based botnet detection system can range from $50,000 to $200,000, with ongoing maintenance costs ranging from $10,000 to
$50,000 per year.

8.3. Technical Challenges

- Scalability: ML-based botnet detection systems must handle large volumes of network traffic in real time, which can be challenging in resource-constrained environments.
- Computational Cost: The high computational cost of training and deploying ML models, especially deep learning models, can be a significant barrier in resource-limited environments.
- Model Complexity: As ML models grow in complexity, they become harder to interpret and debug, making error identification and resolution more challenging.
- Integration: Integrating ML-based detection systems with existing security infrastructure can be complex and time-consuming, particularly in environments with legacy systems.

8.4. Implementation Challenges

- Integration: Integrating ML-based systems with existing security infrastructure can be resource-intensive and technically challenging, especially in environments with legacy systems.
- Configuration: Properly configuring ML-based systems requires significant expertise, which can be a challenge in resource-limited environments.
- Maintenance and Updates: Ensuring proper maintenance and regular updates of ML-based detection systems, including training personnel, can be demanding.

The difficulties associated with detecting botnet attacks through Machine Learning techniques arise primarily from constraints in data, computational costs, and the scalability of the model. Data complications encompass the absence of reputable and tagged datasets and issues surrounding privacy, whereas technical problems emphasize the necessity of developing models that can operate in real-time and in environments with limited resources. In addition, the integration of Machine Learning systems into preexisting systems and guaranteeing the systems' explainability presents another challenge. Solving these obstacles is crucial in order to leverage the full capabilities of Machine Learning in botnet detection and to narrow the distance between research and practical application.

9. Gaps

While the last section highlighted significant issues linked to utilizing ML-based botnet detection systems, this section will attempt to highlight the pending open research issues, which are still unresolved. These gaps are the issues that restrain the effective implementation and use of these systems, thus emphasizing the requirement for different techniques to fully address these issues.

Some of the newer models struggle primarily due to their dependence on labeled data. The volume of relevant labeled data is often scarce, especially when dealing with an emerging threat. Because of this, they tend to achieve comparatively better results with data they have already seen, showcasing a requirement for more diverse and flexible models.

The other problem stems from the fact that training and use of almost all ML models, especially deep models, are very expensive in terms of computational resources. This is a possible hurdle in the adoption of technology, especially when resources are meager. Moreover, barriers such as inadequate transparency and lack of clarity regarding the relationships between different elements of machine learning increase the possibility of not being able to explain the result of a decision, a concern often cherished within sensitive fields.

Along with these technical challenges, there also exists a multitude of operational challenges that need to be solved. For one, the adoption of ML-based botnet detection systems into the security infrastructure already in place tends to be tedious

and costly. In addition, environments with very few resources will find it burdensome to constantly update and maintain the model.

The existing models also fail at obtaining metrics against detection and response to dynamic and multistage threats. A classic example of such models is the ones that fail to detect sophisticated threats' advanced techniques of evasion like code obfuscation or anti-debugging. Apart from that, some models become obsolete due to the quick evolution of threats, which leads to slower detection and response.

Additionally, the lack of benchmarks evaluating the scalability of ML models under real-time constraints remains a critical gap. Current studies rarely address the trade-off between detection accuracy and deployment costs, particularly for IoT devices with limited resources. Exploring lightweight yet high-performing architectures, such as MobileNet variants for deep learning or federated learning for distributed networks, could address these challenges. Furthermore, integrating Explainable AI tools like SHAP or LIME into botnet detection frameworks can bridge the gap between high model accuracy and operational transparency, which is vital for deployment in security-critical environments.

To solve these problems, advancements shall be made in research and development. Their limitations can be dealt with by utilizing transfer learning, synthetic data generation, or adversarial training. Resource-deficient botnet detection models augmented with explainable AI suggest that this is the new frontier to achieve greater efficiency and interpretability.

10. Future Directions

A number of new directions have the potential to disrupt ML-driven botnet detection. The center of focus is steadily shifting to deep learning models. These models portray great promise for detecting advanced persistent threats (APTs) as they are able to effectively recognize complex spatiotemporal patterns of observable network traffic. Sadly, these models are computationally expensive and do not offer much flexibility in terms of accuracy optimization or multi-parameter benchmarking, making their real-world application difficult to justify.

Another important area of focus is the explainability and interpretability of ML models. Sensitive fields such as cybersecurity require trust before a system can be adopted. Making the ML models explainable and usable through SHAP and LIME is useful as it allows business decision-makers to appreciate and confirm the reasoning behind their actions.

Detecting new threats such as botnet malware remains an ongoing problem due to a scarcity of high-quality labeled data. Transfer learning, few-shot learning, and other similar methodologies are beginning to mitigate this problem by allowing models to learn with minimal data. These efforts are particularly useful in cases where large amounts of high-quality data are difficult to acquire.

Additionally, there is an increasing emphasis on integrating ML-based botnet detection systems with broader security infrastructures, such as intrusion detection systems (IDS) and incident response systems. This integration enhances cybersecurity frameworks by adding automated capabilities to detect and respond to threats in real time, significantly improving overall system resilience.

To address the existing difficulties, the following suggestions are administered:

- Create Stronger Generalized Models: Create IoT threat models that work well in extremely low-resource environments.
- Concentrate on Effectiveness and Proportionality: In low-resource environments, real-time deployment requires the use of lightweight scalable architectures.
- Improve Trust and Regulatory Compliance: Design detection models that are actionable and easy to understand, allowing further trust in regulation with ML detection systems.
- Connect to Other Security Systems: The possibility of integrating automated AI-based threat detection systems into the existing security architecture must be explored so that the system's threat management capabilities are properly integrated.
- Outline Maintenance Scheduling: Machine learning models require frequent updates and adjustments to remain useful against new, emerging threats in the field.

Future research should prioritize developing generalized models capable of detecting new threats with minimal retraining, especially in dynamic IoT environments. Leveraging transfer learning, few-shot learning, or synthetic data generation can enhance model adaptability and reduce dependency on large labeled datasets. Another promising avenue is the integration of ML-based botnet detection into broader cybersecurity ecosystems, enabling seamless real-time threat response through interconnected intrusion detection and incident response systems. These advancements could significantly improve both the scalability and operational efficiency of botnet defense mechanisms.

11. Conclusions

In conclusion, the detection and mitigation of sophisticated and evolving threats with the help of machine learning-based botnet detection further prove to be a powerful approach. Significant progress has been achieved so far; however, the difficulty of obtaining labeled information, high compute requirements, and low generalizability remain obstacles. These constraints, especially in the context of resource-limited IoT networks, demand new solutions.

Further research in this area will need to step away from solely building comprehensive models that accommodate all intricacies of data. There will need to be a focus on making these models capable of extrapolating new threats and generalizing to unseen data with limited input. Making these models more efficient while still scalable will ensure practicality when it comes to deploying these systems in resource-deprived environments. Furthermore, the focus on interpretability and

explainability cannot be neglected, as these are imperative to fostering trust and aiding the deployment of machine learning systems in sensitive and regulated fields.

Developments such as deploying deep learning for detection purposes, fusing machine learning-based detection with other security systems, transfer learning, and few-shot learning are highly beneficial. These innovations will fill the gaps that exist and make machine learning-based detection more proactive and efficient in combating the ever-evolving tactics of botnets.

Overall, botnet detection through machine learning techniques presents an incredible opportunity to bolster the effectiveness of the cybersecurity landscape. By closing these existing gaps and capitalizing on these promising trends, organizations will be better equipped to detect, deter, and address incredibly complex real-time cyber threats.

References

[1] S. Kumar, M. Singh, and S. K. Singh, "A survey of machine learning techniques for botnet detection," *Journal of Cyber Security Technology*, vol. 3, pp. 1–15, 2019.

[2] Z. Li, W. Gao, and G. Wang, "A deep learning approach for botnet detection," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2020–2033, 2020. https://doi.org/10.1109/tifs.2020.3003571

[3] I. Santos, J. Devesa, and F. Brezo, "A survey of unsupervised learning techniques for botnet detection," *Journal of Cyber Security Technology*, vol. 3, pp. 1–15, 2019.

[4] N. Koroniotis, N. Moustafa, and J. Slay, "Towards the development of real-time IoT-botnet detection using machine learning," *IEEE Transactions on Industrial Informatics*, vol. 15, pp. 1724–1734, 2019.

[5] N. Moustafa, B. Turnbull, and K. K. R. Choo, "An empirical evaluation of Bot-IoT dataset for network anomaly detection," presented at the IEEE International Conference on Wireless and Mobile Comput, 2019.

[6] W. Gao, Z. Li, and G. Wang, "A survey of machine learning approaches for botnet detection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, pp. 201–214, 2019.

[7] G. Wang and Z. Li, "A survey of machine learning approaches for malware detection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, pp. 201–214, 2020.

[8] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "A multifaceted approach to understanding the botnet phenomenon," presented at the in Proc. 6th ACM SIGCOMM Conf. Internet Meas., 2006, pp. 41–52, 2006.

[9] P. Barford and V. Yegneswaran, "An inside look at botnets. In malware detection." Boston, MA: Springer US, 2007, pp. 171-191.

[10] J. Bakker, "ISCX2012 dataset processing," GitHub," Retrieved: https://github.com/bakkerjarr/ISCX2012DatasetProcessing, 2012.

[11] E. Cooke, F. Jahanian, and D. McPherson, "The zombie roundup: Understanding, detecting, and disrupting botnets," *SRUTI*, vol. 5, pp. 6-6, 2005.

[12] D. Dagon, G. Gu, C. Lee, and W. Lee, "A taxonomy of botnet structures," presented at the in Proc. Annu. Comput. Secur. Appl. Conf, 2007.

[13] J. B. Grizzard, V. Sharma, C. Nunnery, and B. B. Kang, "Peer-to-peer botnets: Overview and case study," in *In Proc. USENIX Workshop Hot Topics Understanding Botnets*, 2007.

[14] J. R. Binkley and S. Singh, "An algorithm for anomaly-based botnet detection," in *In Proc. 2nd Conf. Steps Reducing Unwanted Traffic Internet, 2006, pp. 43–48*, 2006.

[15] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "BotMiner: Clustering analysis of network traffic for protocol- and structure-independent botnet detection," in *In Proc. 17th USENIX Secur. Symp., 2008, pp. 139–154*, 2008.

[16] W. T. Strayer, D. Lapsley, R. Walsh, and C. Livadas, "Botnet detection based on network behavior," in *In Proc. 6th Int. Conf. Appl. Cryptogr. Netw. Secur., 2008, pp. 1–14*, 2008.

[17] S. Garcia, A. Zuniga, and J. Campo, "CTU-13: Thirteen scenarios to evaluate the detection of botnet communication in CTU University," *Journal of Information Security*, vol. 5, pp. 267–284, 2014.

[18] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Computers & Security*, vol. 31, no. 3, pp. 357-374, 2012.

[19] S. Garcia, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Computers & Security*, vol. 45, pp. 100-123, 2014.

[20] C. Tongun, "CTU13 botnet detection," GitHub," Retrieved: https://github.com/tongun/ctu13-botnet-detection. [Accessed 2018.

[21] 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," *arXiv preprint, arXiv:1811.00701*, 2018.

[22] T. Mirza, "Intrusion detection system using deep learning," GitHub," Retrieved: https://github.com/tamimmirza/Intrusion-Detection-System-using-Deep-Learning, 2018.

[23] P. University, "Evaluation of k-means and hierarchical clustering for botnet detection on CTU-13 and ISCX datasets ", Internal Report, 2018.

[24] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and scenario-specific intrusion detection," *IEEE Trans. Inf. Forensics Secur*, vol. 13, pp. 1579–1592, 2018.

[25] 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.

[26] J. Atuhurra, T. Hara, Y. Zhang, M. Sasabe, and S. Kasahara, "Dealing with imbalanced classes in bot-IoT dataset," *arXiv preprint arXiv:2403.18989*, 2024.

[27] U. C. Cyber, "The BoT-IoT dataset," Retrieved: https://www.impactcybertrust.org/dataset_view?idDataset=1296, 2018.

[28] L. Yin, W. Chen, X. Luo, and H. Yang, "Efficient large-scale iot botnet detection through graphsaint-based subgraph sampling and graph isomorphism network," *Mathematics*, vol. 12, no. 9, p. 1315, 2024.

[29] X. Chen, Y. Li, and J. Zhang, "Deep defense: Identifying DDoS attack via deep learning," *IEEE Trans. Netw. Secur*, vol. 45, pp. 123–135, 2023.

[30] T. Kohonen and M. Lehtokangas, "Self-organizing maps in cybersecurity applications: Analyzing botnet traffic on CTU-13," *Cyber Def. Rev*, vol. 12, pp. 45–60, 2024.

[31] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased boosting with categorical features," in *In Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 31, pp. 6638–6648*, 2018.

[32] K. Alissa, T. Alyas, K. Zafar, Q. Abbas, N. Tabassum, and S. Sakib, "Botnet attack detection in IoT using machine learning," *Computational Intelligence and Neuroscience*, vol. 2022, no. 1, p. 4515642, 2022.

[33] M. Beigi, R. Jalili, and M. Amini, "Analyzing ISCX 2012 botnet dataset using a hybrid K-means and Naive Bayes approach," *Journal of Information Security and Applications*, vol. 19, pp. 295–302, 2014.

[34] Y. Wang, J. Zhang, and H. Liu, "SMOTE-DRNN: A deep residual neural network approach for imbalanced Bot-IoT detection," *IEEE Trans. Netw. Secur*, vol. 11, pp. 245–257, 2023.

[35] X. Chen and Q. Li, "Efficient Bot-IoT detection using IDBO-CatBoost: A Bayesian optimized classifier," *Appl. AI Secur*, vol. 7, p. 198, 2023.

[36] R. Patel and M. Desai, "A voting ensemble approach for high-performance Bot-IoT detection," *Int. J. Cybersecur*, vol. 15, pp. 321–334, 2023.

[37] A. Singh, P. Gupta, and R. Verma, "Feature selection with ZOA+DGAN for enhanced Bot-IoT detection," *Journal of Network and Systems Management*, vol. 18, pp. 67–82, 2023.

[38] Gartner, "Market guide for machine learning-based botnet detection," Retrieved: https://www.gartner.com/en/documents/3986225, 2020.