ISSN:2584-2706 DOI: https://doi.org/10.5281/zenodo.21756076

Behavioral Fingerprinting for Rogue Device Detection in IoT Networks: A Comprehensive Review of Generative AI-Driven Approaches

Vedant V. Kokate
Research Scholar P.G Department of Computer Science and Engineering SGBAU Amravati, India

Dr. Swati S. Sherekar
P.G Department of Computer Science and Engineering
SGBAU Amravati, India

Abstract
The rapid proliferation of Internet of Things (IoT) devices across home, office, and network infrastructure ecosystems has introduced significant security challenges, particularly regarding the presence of rogue or unauthorized devices. Due to the dynamic nature of modern networks, traditional security techniques frequently fail to detect these unauthorized assets. To address this, behavioral fingerprinting offers an effective solution for device identification by analyzing network activity, leveraging the fact that different devices exhibit unique communication patterns. This review paper focuses on Generative AI-based approaches for behavioral fingerprinting in IoT networks. Specifically, we examine existing literature on anomaly detection and device identification that utilizes methodologies such as Autoencoders, Variational Autoencoders (VAEs), and Generative Adversarial Networks (GANs). Furthermore, this paper analyzes the performance metrics, computational resource requirements, and security concerns associated with implementing Generative AI within IoT environments.

Keywords: IoT Security, Behavioral Fingerprinting, Generative AI, Rogue Device Detection, Anomaly Detection

I. Introduction

The home, healthcare systems, and industrial networks have adopted Internet of Things (IoT), This adoption of technology transform into network ecosystems. The usage of Internet of Things (IoT) have been significantly increased, it has estimated that 40 billion devices will be connected in networks by the year 2030 [1]. At the same time, this incremental effect has introduced vulnerabilities in network.

Enabling the home to IoT ecosystem, increasing in security vulnerabilities due inconsistent protocols and protection mechanisms, create opportunities for attackers to exploit the networks [1]. Due to the random nature of the IoT, it is difficult to implement reliable and unified security measures in devices manufactured by many suppliers. In addition, these devices are user-friendly specifically designed for technical and non-technical users to manage IoT devices easily [2].

Unauthorized devices mimic valid traffic patterns in IoT networks, were as widely used security techniques such as firewalls, encryption, and signature-based intrusion detection systems often face difficulty to identify threats in networks [3]. Behavioral fingerprinting technique identify device

ISSN:2584-2706 DOI: https://doi.org/10.5281/zenodo.21756076

specific features such as packet timing, communication rates, and protocol by “behavioral signatures” in networks.

Behavioral modeling has advanced with Generative Artificial Intelligence (Generative-AI) which opened new possibilities. Traditional machine learning method have inaccuracy to identify devices in IoT networks, Generative-AI can learn complex multi-dimensional data and generate synthetic behavioral patterns [5]. Behaviors are frequently changing and different than other, these capabilities are valuable for IoT security.

II. Background

This section provides the foundational concepts necessary for understanding Generative AI-driven behavioral fingerprinting for rogue device detection in IoT networks. The background is divided into three parts IoT security challenges, behavioral fingerprinting, and Generative AI models

- **IoT Security Challenges**

The modern infrastructure has been transformed with "Internet of Things" (IoT) by connecting and automating smart homes, medical systems, industrial networks and transportation systems. But this IoT advancement also bring vulnerabilities in IoT environment and not to traditional IT networks.

Beana et al. [1] has point out many challenges that directed to high vulnerability of IoT networks. First, IoT devices have less resources to operate on network, for instance they have limited computational power, memory and battery life. due to this limitation cannot use sophisticated security features such as encryption algorithms or intrusion detection systems. Secondly, different communication protocols such as WiFi, Zigbee, Z-wave, Bluetoothe and LoRa are used in IoT environments which makes it difficult to apply security in a standard way. Third, the changing nature of usage patterns as devices frequently joins and leave networks, creates problem to maintain consistent security policies. Fourth, IoT device often work with sensitive data of users, because of privacy concerns limit the ability to inspect traffic.

Among the most critical security threats in IoT environments is the presence of rogue devices. Rogue devices are unauthorized nodes that either infiltrate the network through compromised credentials or become compromised after legitimate deployment. Andrews et al. [2] differentiate between two main types of rogue device threats: anomalous devices that are not part of the known network estate, and potentially vulnerable devices that are part of the known estate but run outdated software with unpatched vulnerabilities.

Malicious actions taken by rogue devices can include data exfiltration, recruitment of botnets, Distributed Denial of Service (DDoS) attacks and network reconnaissance. The Mirai botnet attack in 2016 showed the devastating potential of compromised IoT networks, using hundreds of thousands of vulnerable devices to launch unprecedented DDoS attacks [6]. Similarly, ransomware attacks targeting healthcare IoT devices have endangered patient safety while demanding substantial payments.

Firewalls, encryption, and signature-based intrusion detection systems are traditional security mechanisms that can be easily bypassed by rogue devices. However, Chowdhury and Abas [3] argue that these mechanisms are not effective as the rogue devices can mimic the legitimate traffic pattern, exploit the zeroday vulnerabilities or use the inherent weaknesses of the resource-constrained devices. In contrast, static identifiers such as MAC addresses and IP addresses are easily spoofed and thus unreliable for device authentication.

- **Behavioral Fingerprinting**

Behavioral fingerprinting is the method to identify IoT devices in network. The technique depends on unique behavioral signature of IoT device, this unique behavioral signature produce by combining the operational features of IoT device.

The differentiating behavioral features are used for "behavioral fingerprinting" such as packet timing patterns, communication frequency, payload sizes, protocol usage, and communication patterns. Because these behavioral patterns are difficult to spoof as compared to static identifiers, this method is more suitable for authentic device identification.

L. Cruz [13] improved the use of behavioral fingerprinting methods by packet inter arrival times and signal fluctuations as the discriminative feature for identify unauthorized devices. Their study shown that deep learning models specifically LSTM networks capture behavioral patterns effectively and

ISSN:2584-2706 DOI: https://doi.org/10.5281/zenodo.21756076

detect unauthorized devices with an accuracy of 94 - 96%. Traditional security mechanisms inefficient to identify rogue device, on other hand behavioral fingerprints offers more advantage. First, the method doesn't require modifications on hardware or software of device's and it is non-invasive. Second, it works passively on network, monitoring network traffic without using additional computational resource. Third, it is resistant to spoofing, as behavioral patterns are linked to the device features. Fourth, zero-day attacks can be detected, since the device behavioral patterns changes far from normal behavior indicates a potential security breach.

Behavioral fingerprinting process has three steps, such as Feature Extraction, Fingerprint Generation and Fingerprint Verification. Feature extraction: Feature extraction deals with identification and extraction of valuable attributes from network flows corresponding to devices. Fingerprint generation: This process will enable in developing a unique identifier that represents genuine behavior of authentic device. Fingerprint verification: In this step, normal and unauthorized activities are supervised.

- **Generative AI Models**

Generative AI is a type of Artificial Intelligence that uses specific trained models to generate new output as per the input prompt. The generative model is a computer program which is trained on complex data and algorithms to helps practice of generative AI. Due to its ability to identify patterns and generate synthetic data it is used in field of IoT security, Autoencoder, Variational Autoencoder, and Generative Adversarial Network these are three models were used for identify anomalies in IoT networks.

- **Autoencoder (AE):**

An autoencoder is a subset of encoder and decoder which is a neural network architecture. Autoencoder do not rely on labeled training data, it is trained to discover hidden patterns in unlabeled data. The data compressed by encoder into a lower-dimensional, while the original input recreated by decoder. Mean Squared Error (MSE) is used to minimize reconstruction error for training on network.

Siwakoti et al. [21] describe an autoencoder architecture designed for anomaly detection in IoT networks. The encoder, with a layer structure of [128, 64, 32], compresses input features into a latent representation, while the decoder, with layers [32, 64, 128], reconstructs the original input data. The model is trained exclusively on legitimate traffic to learn patterns of normal behavior; it is unable to correctly reconstruct patterns it did not encounter during the training process.

- **Variational Autoencoder (VAE):**

A Variational Autoencoder (VAE) builds upon the traditional autoencoder by learning a probability distribution in the latent space instead of a deterministic latent representation. Instead of mapping an input to a point in the latent space, a VAE learns the mean and variance of a probability distribution in the latent space (usually a Gaussian distribution). This probabilistic formulation has the benefit of providing natural regularization for the model, which improves generalization.

Siwakoti et al. [21 proposed a VAE model that has 8 dimensions of latent space where encoder produces the mean and variance vectors and the sample from the distribution is generated in latent space and a weighted loss is used, which consists of reconstruction loss and loss based on KL divergence. It was reported in [5] that using VAE-based models achieves an accuracy of 96.2%, with only 0.51% degradation in F1-score, when trained on synthetic dataset. The benefits of using VAE are natural regularization, which helps in generalizing to previously unseen patterns; capability to generate new data samples from the learned distribution; and possibility to perform anomaly detection by measuring the probability of reconstructing the samples.

- **Generative Adversarial Network (GAN):**

Generative adversarial network The process is comprised of two opposing networks: generator and discriminator that get trained against each other in the adversarial procedure. The generator has the capability of creating artificial data which are similar to the original ones and on the other hand, the discriminator discriminates the real data from the artificially generated data. In their paper [20], the authors Abusitta et al present a GAN framework for the anomaly detection in the field of IoT.

Their generator has 32, 64, 128 units, the synthetic data is being produced using random noise and their discriminator consists of 128, 64, 32 layers.

Through training this architecture for a GAN against each other in a competitive way, the generator is forced to generate more similar real-like data, while the discriminator gets better at detecting synthetic

ISSN:2584-2706 DOI: https://doi.org/10.5281/zenodo.21756076

from real samples. There are many pros when utilizing GANs for securing IoT, including: Adversarial learning ensures GANs are resistant to noisy and varying environments of the IoT sector [20]. The generator is also able to generate artificial traffic samples that can overcome any shortage of data [5]. With the discriminator we can implement anomaly detection as an anomaly generates distinct patterns from real data [20].

III. Related Work

IoT devices fingerprinting using ML approach has a history. Authors Andrews et al. [2] defined 5 main external features of a fingerprinting scheme as training set applicable, testing set distributed, device count, classifier granularity and accuracy, based on which, they propose a Comparative Framework and proposed an effective means of comparison. This work proposes an objective and comprehensive Comparative Framework for fingerprinting based on detailed examination of 69 papers.

Liu et al. [7] presented an intelligent system for device identification called IDENTIFY that generates concise but very distinct features from packet level characteristics. Their approaches utilized feature selection with ranker and information gain, with a Random Forest classifier for device categories and type of device.

Khalid et al. [8] used a Random Forest classifier with 100 decision trees with the Gini impurity index to find out anomalies in IoT network with the help of dataset called RT_IOT2022. For the implementation, we first pre-process the data, then performed feature selection, followed by training a Random Forest model with stratified sampling so as to keep the class proportion the same.

In their study [13], Cruz tested a collection of classification algorithms, namely KNN, SVM, Random Forest, and LSTM neural networks for unauthorized device identification. They proposed to extract a set of features using features such as the packet time, the signal attributes and behavioral activities of device usage patterns. In their model, an LSTM neural network was utilized to discover the ‘time-series’ pattern in the behavioral usage of the device.

LLM-driven intrusion detection with small lightweight LLMs fine-tuned on an IoT data-set (IoT-23, TON_IoT) using supervised learning methods for fine-tuning such as BERT-Small, BERT-Mini and TinyBERT models was implemented by Otoum et al. [25]. The technique involved LLM-based prompting, tokenization, softmax activation function with cross-entropy loss.

Aceto et al. [5] introduced a Conditional Variational Autoencoder (CVAE) for the generation of synthetic traffic. The encoder and decoder of this model had layers of dimensions [128, 64, 32] and [32, 64, 128] respectively. Their approach consisted of the data preprocessing that was implemented through binning, extracting features that consist of packet length, inter-arrival time, TCP window size and packet flags and finally training the model using a loss function composed of cross entropy, MSE and cosine similarity that were summed up.

The work by Abusitta et al. [19] describes a GAN-based anomaly detection system using adversarial training, which utilizes a GAN based auto-encoder, incorporating the discriminator in the loss function in addition to the reconstruction error. They tested this system using DS2OS dataset with 40% noise and the encoder features were fed to standard machine learning classifiers.

Another ensemble model by Siwakoti et al. [20] called AD-GAM, integrates autoencoders, VAEs, and GANs. They performed feature extraction on the CICIoT2023 dataset, then trained each of the autoencoder, VAE and GAN on only normal traffic, and a weighted ensemble voting was carried out. The autoencoder had encoder layers of dimensions [128, 64, 32]. VAE contained a stochastic latent space which was enforced by KL divergence. The GAN comprises a generator and a discriminator.

R. Li et al. [23] developed IoTGemini with a device modeling module and a traffic generation module. The former generated models of functionality and behavior of devices with the aid of a bipartite graph and behavior profiling algorithms. The latter generated traffic with the use of Packet Sequence GAN (PS-GAN) where a unidirectional LSTM and a bidirectional LSTM were employed for the generator and discriminator, respectively to capture the sequence information.

According to Diaf et al. [24], a technique called 'BART Predict' use two models (BART and BERT). The system includes several steps: packet analyzing, extract features at Layers 2, 3 and 4 on each protocol field, the tuning of BART to forecast the next packet and fine tuning of BERT in packet-pair classification. They used 'Byte Pair Encoding' (BPE) for tokenizer on BART, and the model has 6 encoder and 6 decoder layers.

ISSN:2584-2706 DOI: https://doi.org/10.5281/zenodo.21756076

Jeganathan et al. [26] introduce a method called "ShadowScan", which consists of a hybrid method that combine LLMs with RAG. First, network flow features were transformed into natural language prompts to build a 'vector database that store knowledge and issued in an information retrieval process. Then, they fine tune LLaMA, DistilBERT and RoBERTa, where they used features such as SrcPort, DstPort, FlowIATMin, FlowIATMax, FlowDuration and FlowBytes/s.

In short, conventional machine learning techniques are capable of producing significant reduction of the dimension of the data particularly through careful feature selection. Very early in the field of research, such as Liu et al. And Khalid, showed that even static classifiers, such as Random Forest implemented in the tested studies, are able to provide very high detection rates up to 97.4% for identifying networks in structured profiling approaches. However, it must be emphasized that the presented solutions in their papers do view the network data as individual data points or slices and not as continuous traffic.

Consequently, just like all traditional classifiers these algorithms are unable to consider any form of historical or temporal connection context and cannot properly generalize to dynamic and changing network environments. The answer of many subsequent approaches to the problem was the use of frameworks tailored for time series data analysis, marking the genesis of the Deep Learning approaches with sequential models such as the LSTMs discussed in Cruz, that intrinsically accounted for the time dependent behavior of network devices with an average accuracy of 95%. Whilst those sequential models do take advantage of sequentially related data, their successful operation depends on a continuous, balanced, and clean supply of training data which is not the case in real-world conditions and environmental data such as environmental anomalies may lead to degrading performances. To account for the demands of real-world data, generative models came into picture aiming to learn the real distribution of network data, as opposed to classification approaches which aim to classify data points into pre-defined categories, thereby becoming more robust and adaptable to environmental changes. The GAN based autoencoder architecture presented in Abusitta et al., for example, makes use of the discriminator as part of the loss function during training and reports 95.9% detection and 95% F-1 score on a heavily noisy dataset (up to 40%). Although each generative model proves successful individually - such as Autoencoders (92.8% accuracy) and VAEs (93.5% accuracy) - an ensemble of different generative models has demonstrated superior performance - for example 96.2% in Siwakoti et al. With Large Language Models and their inherent ability of tokenization and representation of context within transformers, there is an ability to harness the full semantic content of actual network protocol data for improved detection capabilities, as highlighted in the study by Otoum et al., which has obtained near perfect average accuracy scores up to 99.75% by combining transformers with BERT models.

IV. Material and Methods

The section elaborates on the proposed method using Generative AI for behavioral fingerprinting in IoT networks including the relevant materials and methods, with special emphasis on datasets, features and models. The section comprises three subsections: Datasets, Features Extraction and Generative AI Models with a detailed description and summary in the tables provided below for each section.

A. Datasets for Generative AI in IoT

In the Table I, main datasets used to train and test Generative AI models for IoT security have been represented.

Table I: Summary of Datasets for Generative AI-based IoT Security

Dataset	Devices	Attack Types	Samples	Key Features	Reference
DS2OS	8	7	35,000	IoT communication traces	[19]
IoT-23	23	20	6M	Botnet, DDoS, port scanning	[25]
TON_IoT	31	9	22.3M	Multi-source telemetry	[25]
UNSW IoT Traces	33	7	20M+	Smart home environment	[20]
CICIoT2023	105	33	40M+	Real traffic, diverse attacks	[16]

ISSN:2584-2706 DOI: https://doi.org/10.5281/zenodo.21756076

In **table I** presents the datasets utilized for training and validating the Generative AI-based models for IoT security. These datasets differ from each other in number of devices, the types of attacks they incorporate and sample size. Among these datasets, CICIoT2023, which contains 105 devices and 33 types of attacks, is the largest. In contrast, the smallest dataset among the listed ones is DS2OS which has merely eight devices. In the "Key Feature" column, the main attribute for every dataset is highlighted. And in "Reference" column, where dataset is employed is represented.

B. Feature Extraction for Generative AI

In the Table II, feature extraction used for categorized specific features in Generative AI-based behavioral fingerprinting.

Table II: Feature Categories for Generative AI-based Fingerprinting

Feature Category	Specific Features	Purpose	Papers
Packet-level	**Length, inter-arrival time, direction**	**Pattern recognition**	**[6][13][24]**
Flow-level	**Duration, byte/packet counts**	**Behavioral profiling**	**[20][21]**
Statistical	**Mean, variance, entropy, distribution**	**Anomaly detection**	**[5][21]**
Temporal	**Timing patterns, periodicity, bursts**	**Sequence modeling**	**[24]**

In **table II** presents four feature selection categories for Generative AI-based fingerprinting such as Packet-level, Flow-level, Statistical, and Temporal features. For each category, the specific features are selected for fingerprinting process. The Packet-level feature capture communication patterns, Flow-level features session oriented behavioral profiles, Statistical features detect anomaly, and Temporal features supports sequence modeling. The “Paper” column shows research studies that used for each feature category.

C. Generative AI Model Architectures

In the Table III summarizes the architectures of Generative AI models applied to IoT security.

Table III: Generative AI Model Architectures for IoT Security

Model	Encoder Layers	Decoder Layers	Latent Dim	Activation	Reference
Autoencoder	[128,64,32]	[32,64,128]	16	ReLU	[20]
VAE	[128,64,32]	[32,64,128]	8	ReLU	[20]
GAN	Generator:[32,64,128] Discriminator:[128,64,32]	-	32	Leaky ReLU	[19]
PS-GAN	LSTM-based	LSTM-based	Variable	Tanh	[23]
CVAE	Conditional encoding	Conditional decoding	Variable	ReLU	[5]

In **table III** summarizes the architectures of five generative AI models used for IoT security: Autoencoder, Variational Autoencoder (VAE), Generative Adversarial Network (GAN), Packet Sequence GAN (PS-GAN), and Conditional VAE (CVAE). For each model, the encoder and decoder layer architectures, latent dimension sizes, activation functions, and references are provided. While the layer architectures of the autoencoder and VAE are identical, their latent dimension sizes different. In GANs, separate architectures are used for the generator and the discriminator. PS-GAN employs an LSTM-based architecture for time-series data, whereas CVAE utilizes conditional encoding and decoding. Activation functions such as ReLU and Tanh are selected based on the specific requirements of the model.

ISSN:2584-2706 DOI: https://doi.org/10.5281/zenodo.21756076

V. Analysis and Discussion

This part proposes to explore techniques of using generative AI for designing behavioral fingerprints in IoT networks. This is classified into four parts, i.e., performances of generative AI in classification, performance by attack types, comparisons of methods and the key points of this study.

A. Detection Performance of Generative AI Models

In the Table IV gives the overview of the detection performance by the different Generative AI techniques.

Table IV: Detection Performance of Generative AI Models

Model	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	FPR (%)	Reference
Autoencoder	**CICIoT2023**	**92.8**	**91.2**	**90.5**	**90.8**	**4.2**	**[20]**
VAE	**CICIoT2023**	**93.5**	**92.1**	**91.8**	**91.9**	**3.9**	**[20]**
GAN	**DS2OS**	**95.9**	**94.8**	**95.2**	**95**	**3.1**	**[19]**
CVAE	**Multiple**	**96.2**	**-**	**-**	**95.8**	**-**	**[5]**
AE+VAE+GAN Ensemble	**CICIoT2023**	**96.2**	**95.4**	**95.1**	**95.2**	**2.8**	**[20]**

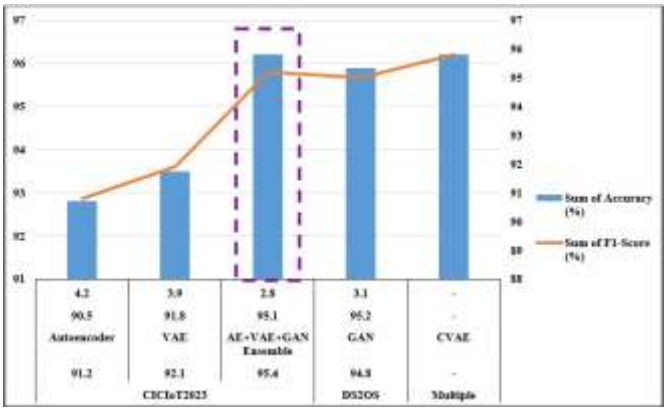

Fig 1: Detection Performance of Generative AI Models

In **Fig 1** By combining models AE, VAE and GAN together in ensemble form, this framework performs the best for all the metrics and showed how beneficial to take advantage of the complementary strengths.

B. Performance by Attack Type

The Table V gives the detection rate for different classes of attacks against Generative AI Ensemble

Table V: Detection Performance by Attack Type

Attack Category	Precision (%)	Recall (%)	F1-Score (%)	Detection Difficulty
DDoS	97.2	96.8	97	Easy
DoS	96.4	95.9	96.1	Easy
Mirai Botnet	96.1	95.4	95.7	Medium
Brute Force	95.1	94.3	94.7	Medium
Reconnaissance	94.8	93.2	94	Medium-Hard
Spoofing	94.2	92.8	93.5	Hard
Web-based	93.5	91.7	92.6	Hard

ISSN:2584-2706 DOI: https://doi.org/10.5281/zenodo.21756076

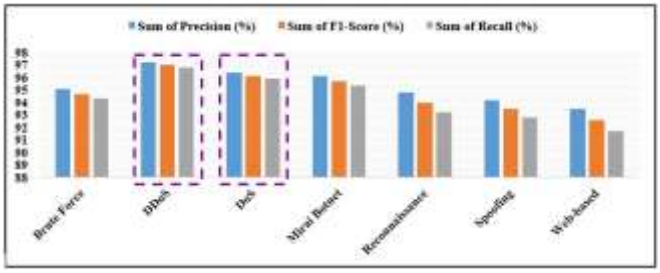

Fig 2: Detection Performance by Attack Type

From **Fig 2** it can be seen that Volumetric attack, such as DDoS or DoS, reaches highest rate of detection because of their well defined patterns whereas web based attacks, in contrast, have to reach relatively lowest rates, due to resemblance with normal traffic patterns.

C. Comparison of Generative AI with Other Approaches

In Table VI comparison between Deep learning, Traditional ML, and Generative AI.

Table VI: Comparison of Generative AI with Other Approaches

Approach	Average Accuracy (%)	Generalization	Edge Suitability	Reference
Random Forest	97.4	Poor	High	[2][7][8]
LSTM	95	Medium	Medium	[13]
BERT	99.75	Medium	Low	[25]
Generative AI (Ensemble)	96.2	Good	Medium	[20]

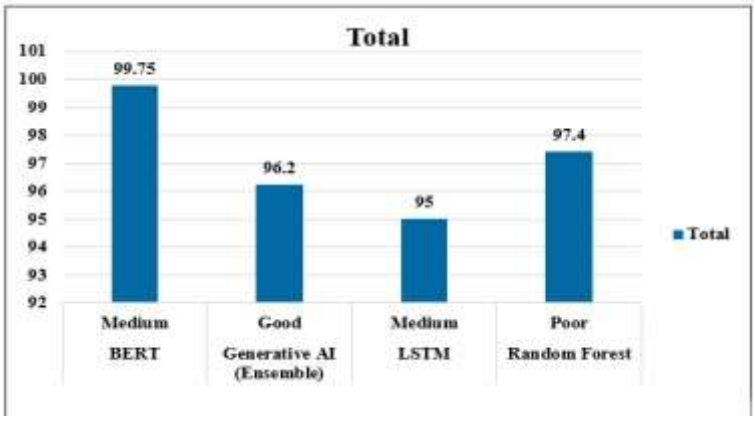

Fig 3: Comparison of Generative AI with Other Approaches

From **Fig 3** it can be seen that Generative AI ensemble model performed significantly well in the generalization ability compared to other traditional approaches, meanwhile has satisfied resources.

D. Result & Analysis

Based on our work on using generative AI for behavioral fingerprinting, we drew the following observations:

1) Ensemble models: Ensembling an autoencoder (AE), Variational autoencoder (VAE), and a generative adversarial network (GAN) based models achieve an accuracy of 96.2%, outperforming

ISSN:2584-2706 DOI: https://doi.org/10.5281/zenodo.21756076

individual models by about 3-5%. The idea behind this ensemble is to make use of the capabilities of each models to detect anomalies, namely the reconstruction ability of AE, the probability based generalization capacity of VAE and its resistance to adversarial attacks, and the GANs to fight against adversarial samples. The combined models were robust against all the attack scenarios tested, while the individual models could be evaded.

2) VAE: The VAE has a natural probability based latent space model which helps it generalize well on unseen device types and patterns [5][20]. VAE based models presented in [5] could generalize to unknown device type and patterns without a significant drop in performances (F1-score dropped only by 0.51%). This property is very useful when dealing with new, frequently appeared devices in the IoT ecosystem.

3) GAN: GAN models can be trained to cope with network disturbances and noisy environment by making use of adversarial training, making them more robust in hostile environment than models based on a de-noising autoencoder [19]. Our GAN model based on [19] reaches a detection rate of 95.9% under 40% noise level.

4) Generative AI: Synthetic traffic generated by GANs and conditional VAEs models can greatly help augment the amount of training data without loss of performance [5] [23]. For example, the F1-score drop when training on synthetic data generated by CVAE is about 0.51%only. The PS-GAN model showed its effectiveness in generating high-fidelity time-series data [23]. Thus, it allows us to overcome the issue of insufficient training data, especially for rare attack events.

5) Autoencoders: Autoencoders and Variational Autoencoders consume less memory (12-14MB), can run on edge CPU, and produce low-detection latencies (2.1 ms–2.4 ms), which would enable us to detect anomalies in real time at the edge rather than relying on the cloud for analysis.

VI. Conclusion

This paper aims to review the Generative AI-enabled techniques of behavioral fingerprinting in IoT Networks. Our study unveils that among all Generative AI Models, especially Autoencoders (AE), Variational Autoencoders (VAE), and Generative Adversarial Networks (GANs), yields greater benefits over the traditional models of behavioral fingerprinting.

The research finding indicates that, in an ensemble learning approach with the combination of AE, VAE and GAN model can get a higher performance for detection rate 96.2% – a set of multiple synergies integrate among generative models can provide better robustness of anomaly detection in comparison with alone the AE, VAE, or GAN. Variational autoencoders has strong generalizability and generative adversarial networks perform great robustness on varying data and varying surrounding conditions. The models can model complex data distribution without supervised labelling of address that the limitation faced in practice during the attack detection using shortage of labeled attack data.

Synthetic traffic generation with Conditional Variational Autoencoders (CVAEs) and Packets Sequence GANs allows to generate highly fidelity synthetic traffic that is capable of maintaining the sequence relationship and behaviors, therefore training models with privacy concern. Though those are promising research directions, more efforts need to be done along these directions to tackle these challenges, which is how the models could generalize well to unknown devices types, optimized for resource-constrained edge device environment.

Finally, the Generative AI methodology of behavior fingerprinting in the area of behavioral fingerprinting offers the next generation smart, flexible, and scalable privacy-preserving security solution against the rogue device threat in IoT systems. Hence, research must be initiated in the areas that are discussed above, in order to narrow the gap between theoretical gains and actual deployment in present day IoT ecosystems.

VII. References

[1] E. Baena, H. Yang, D. Koutsarikolas, and I. Haque, "A Comprehensive Survey on Smart Home IoT Fingerprinting: From Detection to Prevention and Practical Deployment," *arXiv preprint*, vol. 2510.09700, pp. 1-22, 2025.

[2] A. Andrews, G. Oikonomou, S. Armour, P. Thomas, and T. Cattermole, "IoT Device Identification Techniques: A Comparative Analysis for Security Practitioners," *IEEE Access*, vol. 13, pp. 82610-82620, 2025.

ISSN:2584-2706 DOI: https://doi.org/10.5281/zenodo.21756076

[3] R. R. Chowdhury and P. E. Abas, "A survey on device fingerprinting approach for resource-constraint IoT devices: Comparative study and research challenges," *Internet of Things*, vol. 20, p. 100632, 2022.
[4] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A.-R. Sadeghi, and S. Tarkoma, "IoT SENTINEL: Automated device-type identification for security enforcement in IoT," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pp. 2177-2184, 2017.
[5] G. Aceto, F. Giampaolo, C. Guida, S. Izzo, A. Pescape, F. Piccialli, and E. Prezioso, "Synthetic and privacy-preserving traffic trace generation using generative AI models for training Network Intrusion Detection Systems," *Journal of Network and Computer Applications*, vol. 229, p. 103926, 2024.
[6] J.-P. A. Yaacoub, M. Noura, H. N. Noura, O. Salman, E. Yaacoub, R. Couturier, and A. Chehab, "Securing internet of medical things systems: Limitations, issues and recommendations," *Future Generation Computer Systems*, vol. 105, pp. 581-606, 2020.
[7] L. Liu, M. A. Azad, H. Lallie, and H. Atlam, "IDENTIFY: Intelligent device identification using device fingerprints and machine learning," *Pervasive and Mobile Computing*, vol. 114, p. 102103, 2025.
[8] H. Khalid, "Anomaly Detection in IoT Networks Using Machine Learning Techniques," *Dijlah Journal of Engineering Science*, vol. 2, no. 3, pp. 127-136, 2025.
[9] J. Haydar, L. Kanaan, A. Mokdad, and A. Beydoun, "Network Traffic Fingerprinting for IoT Device Identification Using Machine Learning," in *2025 12th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1-7, 2025.
[10] M. Munsif Mir, M. Awrangjeb, W. L. Tan, and A. Zia, "An Improved IoT Device Fingerprinting via Enhanced Feature Selection," in *2024 IEEE International Conference on Communications (ICC)*, 2024.
[11] V. R. Tripathi, S. Tangarala, D. V. Sasanka, D. Reddy, C. S. Dalal, and B. Mitra, "SINTTRA: Sliding Window Based Temporally Aware Network Traffic Analyzer for IoT Device Fingerprinting," in *2024 IEEE 12th International Conference on Intelligent Systems (IS)*, 2024.
[12] L. Tahir, I. Bukhari, K. Munir, H. J. Alyamani, A. Bermak, and A. U. Rehman, "Behavioral Fingerprinting of IoT Devices for Forensic Identification Post-Attack," *IEEE Access*, 2025.
[13] L. Cruz, "Behavioral Fingerprinting Techniques for Rogue Device Detection in Critical IoT Infrastructure Networks," *International Journal of Internet of Things*, vol. 4, no. 1, pp. 1-6, 2026.
[14] O. Hofener and Q. Wang, "Requirements framework for IoT device authentication using behavioral fingerprinting," *Computers & Security*, vol. 154, p. 104459, 2025.
[15] S.-J. Han, D.-H. Shin, S.-S. Yoon, and I.-C. Euom, "Research on Machine Learning-based Packet-level Device Fingerprinting for the Internet of Things," in *2024 IEEE 12th International Conference on Information, Communication and Networks (ICICN)*, pp. 382-387, 2024.
[16] S. Anand, L. Kandregula, T. Szydlo, R. Ranjan, and D. N. Jha, "A Hybrid Monitoring Framework for Anomaly Detection in Smart Home IoT Devices," in *2025 IEEE/ACM 18th International Conference on Utility and Cloud Computing (UCC)*, 2025.
[17] X. Li, Y. Han, and W. Bai, "Analyze the User Behaviors via Network Traffic Data in Smart Home," in *2024 3rd International Conference on Automation, Robotics and Computer Engineering (ICARCE)*, pp. 542-547, 2024.
[18] R. Eda, H. Nakanishi, and N. Togawa, "Contextual Anomalous Behavior Detection in IoT Devices from Power Consumption Using LLM," in *2025 12th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, pp. 37-40, 2025.
[19] A. Abusitta, T. Halabi, A. S. Bataineh, and M. Zulkernine, "Generative Adversarial Networks for Robust Anomaly Detection in Noisy IoT Environments," in *ICC 2024 - IEEE International Conference on Communications*, pp. 4644-4649, 2024.
[20] Y. R. Siwakoti, D. B. Rawat, and S. H. Loke, "AD-GAM: Anomaly Detection in IoT Using Generative AI Models," in *SPIE Future Sensing Technologies*, vol. 1, p. 52, 2025.
[21] J. Wen, J. Nie, J. Kang, D. Niyato, H. Du, Y. Zhang, and M. Guizani, "From Generative AI to Generative Internet of Things: Fundamentals, Framework, and Outlooks," *IEEE Internet of Things Magazine*, vol. 7, no. 3, pp. 30-37, 2024.
[22] H. Xu, Y. Li, O. Balogun, S. Wu, Y. Wang, and Z. Cai, "Security Risks Concerns of Generative AI in the IoT," *IEEE Internet of Things Magazine*, vol. 7, no. 3, pp. 62-68, 2024.

ISSN:2584-2706 DOI: https://doi.org/10.5281/zenodo.21756076

[23] R. Li, Q. Li, Q. Zou, D. Zhao, X. Zeng, Y. Huang, Y. Jiang, F. Lyu, G. Ormazabal, A. Singh, and H. Schulzrinne, "IoTGemini: Modeling IoT Network Behaviors for Synthetic Traffic Generation," *IEEE Transactions on Mobile Computing*, vol. 23, no. 12, pp. 13240-13257, 2024.
[24] A. Diaf, A. A. Korba, N. E. Karabadji, and Y. Ghamri-Doudane, "BARTPredict: Empowering IoT Security with LLM-Driven Cyber Threat Prediction," in *2024 IEEE Global Communications Conference (GLOBECOM)*, pp. 1239-1244, 2024.
[25] Y. Otoum, A. Asad, and A. Nayak, "LLM-Based Threat Detection and Prevention Framework for IoT Ecosystems," *arXiv preprint arXiv:2505.00240*, 2025.
[26] D. Jeganathan, T. H. Bandara, H. S. Gardiyawasam Pussewalage, T. D. Dissanayake, D. S. Fernando, K. S. K. Liyanage, and T. Dahanayaka, "ShadowScan: LLM-Based Device Fingerprinting in IoT Networks," in *2025 IEEE 11th International Conference on Collaboration and Internet Computing (CIC)*, pp. 78-87, 2025.
[27] Y. Shrestha, K. Ansari, and A. Aksoy, "Automated IoT Fingerprinting with LLMs: Harnessing Explainable AI and Artificial Bee Colony Optimization," in *2025 IEEE Security and Privacy Workshops (SPW)*, pp. 184-190, 2025.
[28] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," *Sensors*, vol. 23, no. 13, p. 5941, 2023.
[29] D. Peng, Z. Fu, and J. Wang, "PocketLLM: Enabling On-Device Fine-Tuning for Personalized LLMs," *arXiv preprint*, vol. 2407.01031, pp. 1-12, 2024.
[30] Y. Pimpalkar et al., "Implications of Generative AI on IoT Security: A Dual Perspective," *Advancements in IoT Sensors and Security*, vol. 1, pp. 83-104, 2026.
[31] A. Oacheșu et al., "Enhancing IoT Security with Generative AI: Threat Detection and Countermeasure Design," *Electronics*, vol. 15, issue 1, pp. 92, 2026.