Towards Markov Synthetic BLE Packet Generation for IoT Systems

Katharina O. E. Müller, Keisuke Yokota, Weijie Niu, Daria Schumm, Burkhard Stiller
Communication Systems Group CSG, Department of Informatics IfI, University of Zurich UZH, Switzerland
E-mail:[mueller|niu|shumm|stiller]@ifi.uzh.ch
keisuke.yokota@uzh.ch

***Abstract*—Bluetooth Low Energy (BLE) is widely used in devices like smartphones and personal trackers, but also raises serious privacy risks, especially related to stalking. Machine Learning (ML)-based methods for detecting BLE trackers across vendors show promise, yet are limited by the scarcity and variability of BLE advertisement packets, which hinders model performance. This paper addresses this limitation by introducing the first publicly available, open-source tool for generating synthetic BLE advertisement packets using a Markov model. Designed for structured time-series data, the model can produce all valid BLE packet permutations, addressing a key data gap for research and training. As a case study, synthetic Samsung SmartTag (nearby) packets are used to augment training data, resulting in a 37% increase in median prediction confidence level in real-world evaluations.**

***Index Terms*—Internet of Things, Crowdsourced Finding Networks, Bluetooth Low Energy, Personal Trackers**

I. Introduction

Bluetooth Low Energy (BLE) is a low-power wireless technology widely used to connect Internet of Things (IoT) devices , for example, personal tracking devices, such as Samsung's SmartTag, are being adopted rapidly [1]. While useful for locating lost items, these devices also raise privacy concerns, such as unauthorized tracking [2]. Our recent work has focused on detecting and classifying BLE devices, capturing a 200-hour BLE dataset including all available personal trackers on the market [3] with Machine Learning (ML) classifiers proposed for effective identification [4]. The classifier demonstrated strong performance under controlled conditions, achieving over 99% accuracy on training data. However, this high accuracy did not translate reliably to real-world environments, where only a confidence of 80% was achieved. Further analysis exposed limitations in fine-grained classification tasks, such as identifying different operational states of personal trackers. For instance, distinguishing between a SmartTag in a "lost" state (far from the owner's phone) and one that is "nearby" (within 20 m) proved challenging. The "SmartTag (nearby)" state achieved only $\approx$ 50% accuracy. The issue stems from significant bias within the dataset. A greater number of advertisement packets were collected from AirTags than from SmartTags, due to the limited time the device spent in the nearby state, restricting the classifier's ability to learn representative features. A common challenge, when collecting balanced real-world data.

To address the data imbalance [3] and the limited generalization in BLE packet classification [4], synthetic data generation was explored using Generative Adversarial Networks (GANs) and Markov Models (MMs). GANs were deemed unsuitable due to the few misclassified packets found (≤ 20), which is likely to lead to overfitting or invalid packets. Hence, MMs were chosen for their ability to model structured time-series data, capturing field dependencies and structural constraints inherent in BLE advertisements.

This paper introduces a Proof-of-Concept (PoC) MM-based method that synthesizes high-quality Samsung SmartTag (nearby) packets to augment the training set. The synthetic data improves model robustness across device types and operational states, while laying the foundation for a generalizable, specification-compliant BLE packet generator capable of producing any valid advertisement packet permutation. This paper's contributions are:

1) Open-source mapping of BLE advertisement packets to MM: https://github.com/keyyke/ble_forge
2) Synthetic data to augment the existing dataset
3) Evaluation of effect of synthetic data on real-world classification, demonstrating a $\approx 37\%$ increase in the median model confidence level for SmartTag (nearby)

The paper is organized as follows: Section II introduces background; Section III reviews related work; Section IV presents the synthetic BLE data generation design; Section V details Markov mappings; Section VI evaluates the impact on tracker classification; and Section VII concludes.

II. Background

In BLE communication, devices act as either Central or Peripheral [16], [17]. Smartphones typically serve as Centrals, while personal trackers act as Peripherals. Peripherals broadcast advertisement packets to announce their presence, allowing nearby Centrals to initiate connections. Once connected, the Peripheral stops advertising and enters one-to-one communication [17]. These pre-connection advertisement packets can be analyzed to identify and classify Peripheral devices. The collected data contains only metadata and encrypted payloads for secure pairing, with no personal information such as location or activity, and thus does not compromise user privacy.

A. BLE Packet Structure

Packets consist of a Preamble, Access Address, Protocol Data Unit (PDU), and Cyclic Redundancy Check (CRC) (*cf.* Figure 1). Since advertising occurs via the PDU, this component is the primary focus. There are two PDU types: Advertising Physical Channel PDUs (used for broadcasting) and Data Physical Channel PDUs (used for Central device communication), though only the former is relevant here. The payload structure varies slightly by PDU type.

The PDU comprises a Header and a Payload containing the Advertising Address (AdvA) and Advertising Data

978-3-903176-75-1

TABLE I: Overview of the Related Work

Research	Synthesizing BLE Packet Data	Related		Hidden Markov Model	Time Series Data	Can It Be Used to Create Synthetic BLE Packets?	
		Synthesizing Data	Simulation			Method	Data
[5], [6]	No	Yes	No	No	Yes	No	Partially
[7], [8]	No	No	Yes	No	Yes	No	Partially
[9]	No	Yes	No	No	Yes	No	Partially
[10], [11], [12]	No	Yes	No	No	Yes	Yes	No
[13]	No	Yes	No	Yes	Yes	No	No
[14], [15]	No	Yes	No	Yes	No	No	No
This Paper	Yes	Yes	No	No	Yes	Yes	Yes

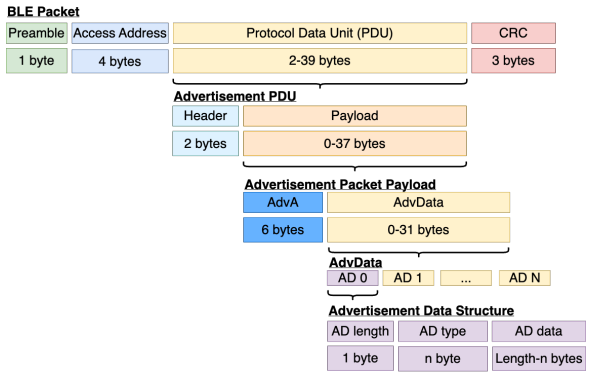


Fig. 1: BLE Advertisement Packets Structure adapted: [17]

(AdvData). AdvA identifies the transmitting device, though BLE devices may use multiple or rotating addresses for privacy. AdvData includes fields such as device name and manufacturer identifier. AdvData is further structured into AD length, AD type, and AD data, which represent the total size, data type, and actual content, respectively.

B. Synthetic Data and MM

Synthetic data replicates the structure and statistics of real data while reducing cost and effort, and offers benefits such as privacy, scalability, and bias reduction via augmentation [18]. It can be generated through statistical methods, ML, or deep learning [19]. Here, the statistical approach with an MM is used. A stochastic process with the memoryless property: the next state depends only on the current one. For instance, a cloudy state may transition to rainy (50%), sunny (20%), or stay cloudy (30%). This property makes MMs suitable for structured data such as BLE packets [20].

III. Related Work

This section reviews related work on BLE and Markov Models (MM) for synthetic time series generation, noting that none address synthetic BLE packet data.

A. Synthetic Data Generation for BLE

For example, [5], [6] generated synthetic RSSI data for BLE location tracking using the Wasserstein interpolation method. However, this focused only on signal strength rather than packet data. [7] simulated BLE advertising packet collisions, and [8] developed a MATLAB Simulink library for BLE sensor network simulation. Moreover, synthetic data has been generated for IP packets and TCP/UDP flows. For instance, [9] created synthetic network traffic data to support the development of intrusion detection systems. These studies addressed privacy protection and the high cost of data collection, but they did not focus on BLE packets.

B. Creation of Synthetic Data Using Markov Models

[10] used an MM to generate high-resolution synthetic solar radiation data by modeling the clarity index time series. [11] applied a first- and second-order Markov model to create 5-minute wind speed data from 30-minute intervals, and [12] generated synthetic wind power outputs. All studies produced data statistically similar to real observations. Hidden MM have also been used. [13] generated financial time series data, [14] created disease incidence data, and [15] modeled population data, capturing complex attribute dependencies such as age, gender, and education. These studies show that both MM and HMM are suited for time series data. MMs work for straightforward patterns such as wind speed, while HMMs handle more complex state transitions, such as in health or demographic data. However, no study has used either to generate synthetic BLE packets.

IV. Dataset and Synthetic Data Generation

This paper builds on our prior work [3], [4].

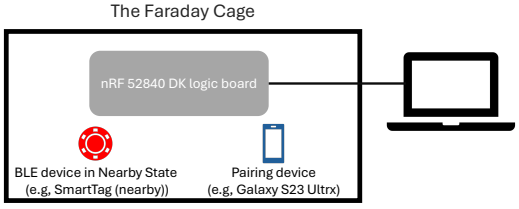


Fig. 3: Faraday Cage Setup: Samsung Nearby State [3]

[3] collected 30 million BLE packets in a controlled environment using a Faraday cage to eliminate external interference. An nRF 52840 DK logic board captured packets from various BLE devices, including but not limited to: AirTags, SmartTags, iPhones, MacBooks, and Lenovo laptops. For "nearby" state data, both the tracker and its paired device (*cf.* Figure 3) were included. The controlled setup ensured clean, labeled data, as shown in Table II. Additionally, unlabeled data was collected at a central train station. To improve real-world classifier confidence, increasing the quantity and diversity of training data helps models better handle noise and interference. Although enhancing feature extraction or model complexity is an option, [4] suggests the existing features are sufficient and the model already performs well on test data, so further tuning will not yield significant gains. Thus, the primary bottleneck is the lack of balanced real-world data, motivating the generation of synthetic data.

TABLE II: Example of the Dataset Created by [3] (SmartTag: Lost)

Number	Time	Source	Destination	Protocol	Channel	Length Packet	Length Header	AD Type	Company ID	Service Data	PDU
1	1712778478	6d:40:e6:f1:49:01	ff:ff:ff:ff:ff:ff	LE LL	37	63	37	Flags, 16-bit Service Class UUIDs (incomplete), Service Data - 16-bit UUID	Samsung Electronics Co., Ltd.	130f050217cfe bfab1c2e0deb7 000000558045e8	ADV_IND
2	1712778480	6d:40:e6:f1:49:01	ff:ff:ff:ff:ff:ff	LE LL	38	63	37	Flags, 16-bit Service Class UUIDs (incomplete), Service Data - 16-bit UUID	Samsung Electronics Co., Ltd.	130f050217cfe bfab1c2e0deb7 000000558045e8	ADV_IND
…	…	…	…	…	…	…	…	…	…	…	…

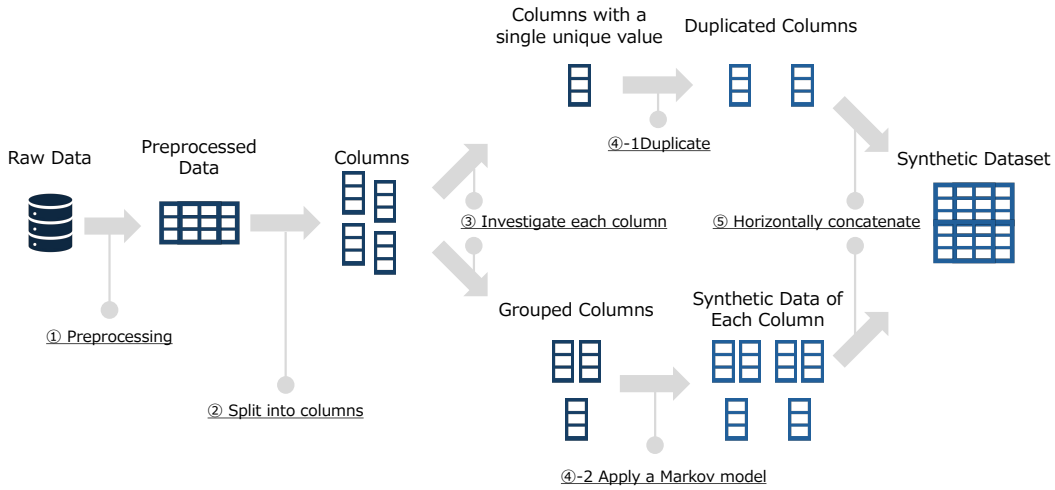


Fig. 2: Pipeline for Synthetic BLE Packet Data Generation

Figure 2 shows the synthetic data creation pipeline. The raw data was first preprocessed and then separated into individual columns for analysis. The columns that contained only a single unique value (0), such as AD Tx Power Level and UUID Tile, were therefore not processed using an MM; instead, synthetic data was generated for them by directly duplicating the original values, not requiring unique models. The remaining columns, with two or more unique values and high correlation, were grouped and treated as single-state transition units:

- Length Packet, Length Header, Length MS Data, Length Service Data: All define packet length.
- CH 37, CH38, CH 39: Define used channel.
- AD Flags, AD Service Data 16-bit UUID, AD 16-bit Service Class UUIDs (incomplete), AD Other: All define AD type.
- UUID Samsung, UUID Other: All define the UUIDs.
- PDU ADV_IND, PDU Other: All define PDU type.
- ST 5, ST Other: All define SmartTag type.

These grouped state models represent the key adaptation of MMs and the paper's main contribution. For each group, transition probabilities were derived to generate synthetic data, which was then combined into a dataset of 600,000 rows and 30 columns. To avoid bias from initial state selection, all possible states were treated as starting points with balanced sampling, ensuring even rare states were represented. The synthetic data was evaluated by (1) comparing value distributions between original and synthetic data and (2) comparing model performance when trained on original versus mixed data.

V. Results: MM State Transition Models

This section describes each grouped state transition model and the probability matrices.

A. Length (Packet, Header, MS Data, Service Data)

This group defines the various observed field length values, indicating valid and correctly structured Samsung SmartTag (nearby) packets. A valid packet requires the Packet Length to exceed the Header Length by at least 26 bytes. Based on this, the valid (Packet Length, Header Length) combinations are (63, 37), (38, 12), and (32, 6). The combination (36, 37), observed only 11 times out of 24,038 data points, appears to deviate from this rule and suggests an unusual packet structure. Notably, Packet Length 63 occurred 23,837 times, while lengths 38 and 32 appeared 95 times each. MS Data Length of 176 appeared only twice. These rare values may reflect manufacturer-specific behavior between the Galaxy smartphone and SmartTag during data collection. While the exact cause is unclear, synthetic data was generated to include these variations.

Figure 4 shows the transition diagram for the Packet Length Group, illustrating the state transition probabilities between different length configurations, including Packet, Header, MS Data, and Service Data Length, from left to right. For example, 63_37_0_160 means that the Length of the Packet is 63, the Length of the Header is 37, the Length of MS Data is 0, and the Length of Service Data is 160. Showing the transition probability is heavily concentrated on specific states, particularly 63_37_0_160.

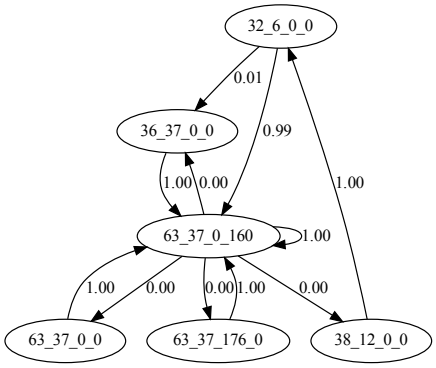


Fig. 4: State Transition Model for Packet Length

B. Chanel: 37, 38, or 39

SmartTag (nearby) consistently operated on a single channel at any given time, with no instances of simultaneous or absent channel use. Channel 37 was used most frequently, with transition probabilities exceeding 60% from channels 38 or 39, and a self-transition probability above 50%. Channel 38 was the second most used, followed by 39. The transition probabilities, summarized in Table III, confirm Channel 37 as the dominant state and highlight the high likelihood of transitions from Channel 38 to 37.

TABLE III: Transition Probability Matrix: Channel

	CH 39	CH 38	CH 37
CH 39	0.000000	0.000419	0.999581
CH 38	0.333054	0.001398	0.665549
CH 37	0.000069	0.492758	0.507173

C. Advertising Data Type

SmartTag (nearby) used three types of Advertising Data: AD Flags, AD Service Data (16-bit UUID), and AD 16-bit Service Class UUIDs. Advertising Data other than these occurred very rarely (5 times out of 24,038). Figure 5 shows the state transition diagram for Advertising Data. It illustrates that AD Flags, AD Service Data (16-bit UUID), and AD 16-bit Service Class UUIDs commonly appear together. In contrast, other advertising data combinations are rare, as reflected by their low transition probabilities.

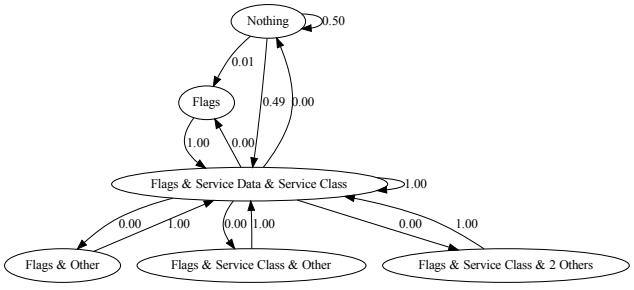


Fig. 5: State Transition Model: Advertising Data Type

D. UUID Type

SmartTag (nearby) typically stored two identical Samsung UUIDs, such as "Samsung Electronics Co., Ltd., Samsung Electronics Co., Ltd.," labeled as "2 Samsung" in Table IV. In one rare case, only one Samsung UUID was present. Additionally, non-Samsung UUIDs appeared only four times and are labeled as "Other." Although the reason for their presence in data from a Samsung device is unclear, these rare cases were included in the synthetic data to preserve statistical realism. Table IV confirms that "2 Samsung" is the dominant state. Additionally, transitions from "Nothing" tend to lead to "2 Samsung" with a probability of approximately 52%, reflecting the prevalence of this configuration.

TABLE IV: Transition Probability Matrix: UUID Type

	Nothing	Other	Samsung	Samsung & Other	2 Samsung
Nothing	0.475248	0.000000	0.000000	0.000000	0.524752
Other	0.000000	0.000000	0.000000	0.000000	1.000000
Samsung	0.000000	0.000000	0.000000	0.000000	1.000000
Samsung&Other	0.000000	0.000000	0.000000	0.000000	1.000000
2 Samsung	0.004448	0.000084	0.000084	0.000084	0.995300

E. PDU Type

Most SmartTag (nearby) PDU types are ADV_IND, indicating that the device is advertising and ready to accept connections from a Central. Other PDU types, primarily SCAN_REQ, likely originate from the paired smartphone in the Faraday cage. As shown in the transition probabilities, ADV_IND dominates with a self-transition rate of approximately 99.6%, while "Other" types occur infrequently and transition equally between themselves and ADV_IND.

F. SmartTag Type

SmartTags in the "nearby" state predominantly use SmartTag Type 5 (ST 5), making it the most common transition state. However, instances with missing or alternative types were also observed, likely originating from the paired smartphone. Table V shows that ST 5 is the dominant state, with a self-transition probability of approximately 99.5%. Transitions from "Nothing" to ST 5 occur about 53% of the time, further confirming its prevalence. "Other" types are rare and transition exclusively to ST 5.

TABLE V: Transition Probability Matrix: SmartTag Type

	Nothing	Other	ST 5
Nothing	0.466019	0.000000	0.533981
Other	0.000000	0.000000	1.000000
ST 5	0.004616	0.000042	0.995342

VI. Evaluation of Synthetic Data

This section evaluates the synthetic data by analyzing its structure and its impact on classification performance. The level of detail is necessary to ensure the data is not only accurate but also effective for its intended use.

A. Comparison of Data Structures

Comparisons of data structures were made for each of the MM state transition models.

1) Length (Packet, Header, MS Data, Service Data): Table VI provides the numerical breakdown of the proportions of different length combinations (Packet, Header, MS Data, and Service Data) between the original dataset and the synthetic dataset. The values confirm that the structure of the data and the most frequently occurring length combination (63_37_0_160) are the same in both datasets.

TABLE VI: Length Distribution Comparison

	32_6_0_0	36_37_0_0	38_12_0_0	63_37_0_0	63_37_0_160	63_37_176_0
Original Data	0.395208	0.045761	0.395208	0.012480	99.143024	0.008320
Synthetic Data	0.403333	0.049167	0.403167	0.007167	99.131000	0.006167

2) Channel: No significant structural change concerning channels exists, with CH37 being the most common, followed by CH38 and CH39, seen in Table VII. The values indicate that the relative distribution of channels remains nearly identical between the original and synthetic datasets, with only minor variations in CH 38 and CH 39.

TABLE VII: Channel Distribution Comparison

	CH 37	CH 38	CH 39
Original Data	60.316998	29.765371	9.917630
Synthetic Data	60.334167	29.711667	9.95416

3) Advertising Data Type: As shown in Table VIII, there is no significant structural change in the Advertising data type between the original and synthetic data. Flags, Service Data, and Service Class Advertisement Data are often used simultaneously, with the most frequently occurring advertising data type remaining consistent across both datasets, with minor variations in the lower-frequency categories.

TABLE VIII: Advertising Data Type Distribution Comparison

	Nothing	Flags	Flags & Other	Flags & Service Class & Other	Flags & Service Class & 2 Others	Flags & Service Data & Service Class
Original Data	0.790415	0.045761	0.004160	0.012480	0.004160	99.143024
Synthetic Data	0.821167	0.047167	0.002167	0.010167	0.001167	99.118167

4) UUID Type: Table IX provides the numerical breakdown of these proportions, confirming there is no significant structural change in UUID type between the original and synthetic data. In both cases, UUIDs in which the name Samsung appears twice are primarily used, such as "Samsung Electronics Co., Ltd., Samsung Electronics Co., Ltd.", indicating that "2 Samsung" is the dominant UUID type.

TABLE IX: UUID Type Distribution Comparison

	Nothing	Other	Samsung	Samsung & Other	2 Samsung
Original Data	0.840336	0.008320	0.008320	0.008320	99.134703
Synthetic Data	0.847667	0.008500	0.006833	0.005167	99.131833

5) PDU Type: Table X provides the numerical breakdown of these proportions, showing no significant structural change in PDU type between the original and synthetic data. In both cases, ADV_IND accounts for more than 99% of the data.

TABLE X: PDU Type Distribution Comparison

	PDU ADV_IND	PDU Other
Original Data	99.209585	0.790415
Synthetic Data	99.219500	0.780500

6) SmartTag Type: Table XI presents the distribution of SmartTag types in both the original and synthetic datasets. ST 5 consistently accounts for more than 99% of the data in both cases, indicating no significant structural differences. Minor variations were observed, with some data having no SmartTag type at all.

TABLE XI: SmartTag Type Distribution Comparison

	ST 5	ST Other	ST Nothing
Original Data	99.138863	0.004160	0.856976
Synthetic Data	99.134167	0.003167	0.862667

7) Summary: The comparison showed no significant structural differences between the synthetic and original datasets, confirming that the MM-based generation preserves data structure. Although initializing from all possible states could introduce deviations, the results indicate that the method avoids malformed packets and maintains BLE protocol compliance, crucial for classification tasks.

B. Model Accuracy and Confidence Comparison

In this section, we adopt the neural network from [4] to assess the impact of synthetic data. The model was trained (1) on original data only, (2) on a mix of original and synthetic data, and (3) with varying amounts of synthetic data to identify the threshold needed for significant gains in real-world classification. Performance was evaluated using confusion matrices and model confidence on the test data.

1) Training Datasets: The original dataset contained 8,734,048 BLE advertisement packets, from which 240,380 were selected for training. Adding MM-generated synthetic data increased this to 328,290. To balance training, all classes were aligned to the size of the smallest class, originally SmartTag (nearby) (Table XII), leading to downsampling of others. After generating 600,000 synthetic samples for SmartTag (nearby), SmartTag (lost) became the new smallest class, and all labels were resampled to this new minimum, yielding a larger balanced training set.

TABLE XII: Samples per label in original, training, and MM-augmented training datasets

Label	**Original Dataset**	**Training Dataset**	**Training + MM Synthetic**
iDevice	5,775,063	24,038	32,829
other Device	1,776,914	24,038	32,829
FindMy Tracker (unpaired)	304,646	24,038	32,829
FindMy Tracker (lost)	191,815	24,038	32,829
FindMy Tracker (nearby)	190,271	24,038	32,829
iDevice FindMy online	189,946	24,038	32,829
iDevice FindMy offline	182,397	24,038	32,829
Tile (lost)	66,129	24,038	32,829
SmartTag (lost)	32,829	24,038	32,829
SmartTag (nearby)	24,038	24,038	32,829

2) Model: The neural network leverages the Softmax function for confidence scoring, while offering the capability in learning nonlinear patterns and distinguish subtle differences between real and synthetic data. Therefore, the neural network was selected for evaluating the synthetic dataset. The model used follows the MLP classifier implementation from the scikit-learn library, with one hidden layer of 100 ReLU-activated neurons and a Softmax-activated output layer for class probability estimation. For reference, the model trained exclusively on original data is denoted as Model-O, while the model trained with both original and MM-generated synthetic data is referred to as Model-M.

3) Classification Accuracy Comparison: Model-O already achieved a high overall accuracy, especially for common and well-represented classes such as iDevice, Tile (lost), SmartTag (lost), and FindMy Tracker (unpaired), all achieving 100% accuracy. With only slight performance drops for underrepresented classes such as SmarTag (nearby) with 99.1% accuracy and 0.9% leakage into the Other Devices class, and more variable classes, such as FindMy Tracker (nearby) with 98.3% accuracy and the remaining 1.7% misclassified as "lost".

Model-M maintained the 100% accuracy for all common and well-represented classes. With slight but notable improvements to the FindMy Tracker (nearby) class, with accuracy improved to 98.5% (from 98.3%) and SmartTag (nearby) demonstrating an improvement to 99.9% accuracy (from 99.1%), and therefore a 0.8% reduction in misclassification as Other Device as summarized in Table XIII.

TABLE XIII: Comparison of Classification Accuracy Between Model-O and Model-M

Class	Model-O Accuracy	Model-M Accuracy	Gain
SmartTag (nearby)	99.1%	99.9%	+0.8%
FindMy Tracker (nearby)	98.3%	98.5%	+0.2%
Other major classes	100.0%	100.0%	—

Therefore, Model-M enhances accuracy on underrepresented classes, with no degradation on well-represented ones. This shows that MM-generated synthetic data improves class balance and generalization, especially for edge cases like SmartTag (nearby). To rule out data size as the cause of performance differences, the model was trained on varying dataset sizes, from 1/64 up to the full training set. Accuracy remained near 100% across all sizes, suggesting that the observed improvement is attributable to the inclusion of synthetic data rather than dataset size.

4) Confidence Level: Accuracies are compared only on the labeled test set, where ground truth is available. For real-world inference data without labels, performance is evaluated via prediction confidence, defined as the maximum class probability assigned by the model for each input. To compare Model-O and Model-M, confidence scores are ranked and plotted by percentile, showing how certainty varies across predictions. Applied to the $Bahnhof\_v2$ dataset, Model-O's confidence for SmartTag (nearby) plateaus near 62%, while Model-M remains close to 100% across most packets, dropping only at the tail (Figure 6a). Thus, incorporating MM-generated synthetic data raises median confidence by $\approx 37\%$, indicating improved generalization and more reliable predictions under real-world conditions.

Model-M improves confidence not only for SmartTag (nearby) but also for FindMy Tracker (lost/nearby), though confidence decreases slightly for FindMy Tracker (unpaired). For Tile (lost) and SmartTag (lost), both models already achieve near-100% confidence with minimal decline, as these classes were well-represented and structurally consistent. While no major gains were expected there, importantly, synthetic data did not degrade performance, confirming that MM-based augmentation is robust and structurally aligned even when improvement is unnecessary.

For the Other Device class, which contain greater variability and noise, synthetic data produced a moderate gain. Model-M's confidence declined slightly earlier than Model-O's but tapered more smoothly, with a higher minimum (40% vs. 30%). Indicating improved generalization across diverse inputs, showing that MM-generated synthetic data boosts performance for underrepresented or complex classes while preserving reliability for well-performing ones.

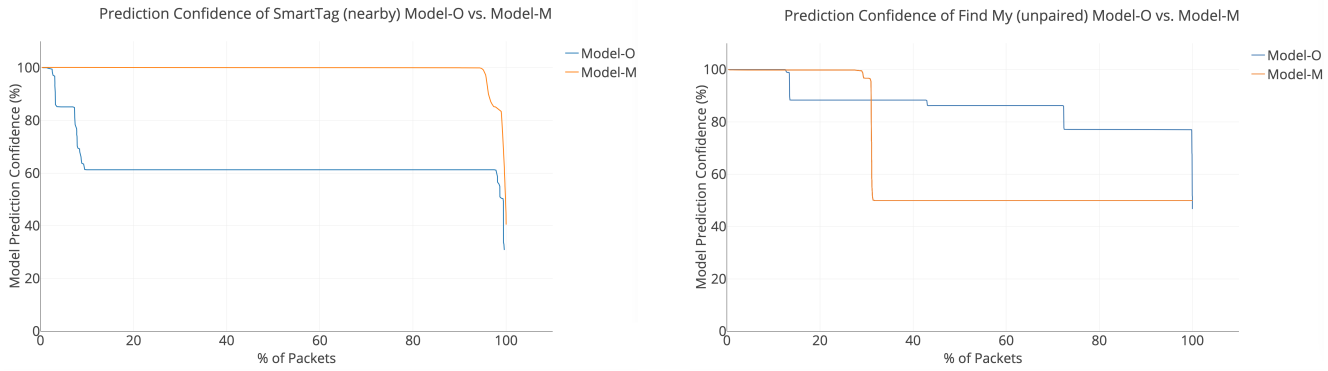
However, the FindMy Tracker (unpaired), see Figure 6b, Model-O (blue), begins at 100% confidence and maintains high certainty of $\approx 90\%$ up to around 40% of the packet distribution. It then undergoes several small declines, stabilizing at a plateau of 88%, followed by a steeper drop beginning after the 80th percentile, eventually reaching a tail confidence of roughly 47%. In contrast, Model-M (orange), which is trained with additional synthetic data generated, also starts near 100% confidence but shows a sharp decline much earlier, around the 30th percentile. The confidence then levels off at a significantly lower plateau of $\approx 50\%$, with minimal further variation. This suggests that the synthetic data introduced greater variability, lowering the model's overall confidence. Unlike most other classes, Model-M underperforms relative to Model-O in this case, indicating that the synthetic data may not have adequately captured the distributional characteristics of FindMy Tracker (unpaired), thereby reducing model certainty.

C. Impact of Synthetic Data Volume on Model

To evaluate the impact of synthetic data, the ratio of synthetic to real training samples for the SmartTag (nearby) class was systematically varied, while keeping the total number of samples per label constant at 24,038 (as shown in Table XII). The proportion of synthetic data was incrementally adjusted to 0%, 1%, 5%, 10%, 30%, 50%, 75%, and 100%. No substantial changes in model performance were observed up to the 50% synthetic data. However, at 75% and 100%, an overall increase in confidence levels was noted, particularly for SmartTag (nearby), where the model demonstrated exceptionally high confidence when trained entirely on synthetic data. For all other device classes, the variation in synthetic data proportion had no significant effect on confidence levels.

D. Discussion and Limitations

MM-generated synthetic data improved classification by increasing diversity rather than volume, preserving individual feature distributions while introducing more unique combinations that enhanced generalization to real-world inputs. However, several risks and limitations must be considered. Overfitting to synthetic patterns is possible: while most classes benefited, FindMy Tracker (unpaired) showed decreased confidence, likely due to altered feature correlations, underscoring the risk of mismatches between synthetic and real-world dependencies. Generalizability is also limited, as our study focused on SmartTag (nearby), and devices with different communication patterns (e.g., Tile, AirTag) may behave differently. Moreover, the MM's memoryless assumption may fail to capture longer-term dependencies such as interference patterns or battery-related effects. Finally, although grouping related fields reduced the risk of implausible feature combinations, independently



(a) Model Confidence Comparison: SmartTag (nearby)

(b) Model Confidence Comparison: Find My (unpaired)

Fig. 6: Changes in Models Predicition Confidence from Model-O to Model-M

modeling column groups can still yield invalid states, even if rare in our evaluation. These limitations highlight that MM-based augmentation is promising and effective in many cases, but not universally reliable, requiring careful validation before use in critical applications.

VII. Summary and Future Work

This work addresses three key limitations: the scarcity of labeled BLE advertisement data, especially for underrepresented classes; the resulting lack of generalization in real-world environments; and the high cost and privacy risks of collecting balanced datasets.

To address these issues, an MM-based method for generating realistic, structurally valid synthetic BLE packets was presented. It augmented the dataset without introducing errors, as confirmed by detailed structural comparisons, and evaluation showed significant gains in model performance, boosting the median model confidence in SmartTag (nearby) classification by 37% and training model accuracy from 99.1% to 99.9%. Crucially, these improvements stemmed from increased data diversity rather than volume, enhancing real-world generalization. Importantly, performance on well-represented classes remained unchanged, confirming its safe integration into the training process.

Future work will explore extending this approach to other personal tracker types and to raw BLE data beyond preprocessed fields, paving the way toward general-purpose BLE data synthesis for robust IoT classification that can be leveraged for any future IoT system.

References

[1] Mordor Intelligence. (2024) Smart tracker market size & share analysis - growth trends & forecasts (2024 - 2029). Mordor Intelligence. [Accessed: 23.07.24]. [Online]. Available: https://www.mordorintelligence.com/industry-reports/smart-tracker-market

[2] T. Yu, J. Henderson, A. Tiu, and T. Haines, "Security and privacy analysis of samsung's crowd-sourced bluetooth location tracking system," in *USENIX Security Symposium*, 2024. [Online]. Available: https://www.usenix.org/conference/usenixsecurity24/presentation/yu-tingfeng

[3] K. O. Müller, S. Saxer, D. Schumm, W. Niu, B. Rodrigues, and B. Stiller, "AirTagged: A Dataset and Task-Group-Framework for Heterogenous High-Density IoT Environments," 2025, manuscript submitted for publication.

[4] K. O.E. Müller, S. Saxer, D. Schumm, B. Rodrigues, B. Stiller, "Tracking Trackers: ML-based Detection in Crowded IoT Environments," 2025, manuscript submitted for publication.

[5] F. S. Daníş, A. T. Cemgíl, and C. Ersoy, "Adaptive sequential monte carlo filter for indoor positioning and tracking with bluetooth low energy beacons," *IEEE Access*, vol. 9, pp. 37 022–37 038, 2021.

[6] F. S. Danış and A. T. Cemgil, "Model-based localization and tracking using bluetooth low-energy beacons," *Sensors*, vol. 17, no. 11, 2017. [Online]. Available: https://www.mdpi.com/1424-8220/17/11/2484

[7] M. Ghamari, E. Villeneuve, C. Soltanpur, J. Khangosstar, B. Janko, R. S. Sherratt, and W. Harwin, "Detailed examination of a packet collision model for bluetooth low energy advertising mode," *IEEE Access*, vol. 6, pp. 46 066–46 073, 2018.

[8] R. Shavelis and K. Ozols, "Bluetooth low energy wireless sensor network library in matlab simulink," *J. Sens. Actuator Networks*, vol. 9, p. 38, 2020.

[9] G. Noblet, C. Lefebvre, P. Owezarski, and W. Ritchie, "Netglyph: Representation learning to generate network traffic with transformers," in *2024 20th International Conference on Network and Service Management (CNSM)*, 2024.

[10] B. Ngoko, H. Sugihara, and T. Funaki, "Synthetic generation of high temporal resolution solar radiation data using markov models," *Solar Energy*, vol. 103, pp. 160–170, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0038092X14001042

[11] Z. Wang and J. Olivier, "Synthetic high-resolution wind data generation based on markov model," in *2021 13th IEEE PES Asia Pacific Power Energy Engineering Conference (APPEEC)*, 2021, pp. 1–6.

[12] K. Brokish and J. Kirtley, "Pitfalls of modeling wind power using markov chains," in *2009 IEEE/PES Power Systems Conference and Exposition*, 2009, pp. 1–6.

[13] P. Nystrup, H. Madsen, and E. Lindström, "Stylised facts of financial time series and hidden markov models in continuous time," *Quantitative Finance*, vol. 15, pp. 1531 – 1541, 2015.

[14] P. Green and S. Richardson, "Hidden markov models and disease mapping," *Journal of the American Statistical Association*, vol. 97, pp. 1055 – 1070, 2002.

[15] I. Saadi, A. Mustafa, J. Teller, B. Farooq, and M. Cools, "Hidden markov model-based population synthesis," *Transportation Research Part B: Methodological*, vol. 90, pp. 1–21, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0191261515300904

[16] C. S. W. Group, "Bluetooth core specification v5.4," Bluetooth Special Interest Group, Tech. Rep., January 2023, accessed: 11-09-2024. [Online]. Available: https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=556599

[17] N. Semiconductor, "Bluetooth LE Advertising," https://academy.nordicsemi.com/courses/bluetooth-low-energy-fundamentals/lessons/lesson-2-bluetooth-le-advertising/topic/advertisement-packet/, 2024, accessed: 1-09-2024.

[18] A. Bauer, S. Trapp, M. Stenger, R. Leppich, S. Kounev, M. Leznik, K. Chard, and I. Foster, "Comprehensive exploration of synthetic data generation: A survey," 2024. [Online]. Available: https://arxiv.org/abs/2401.02524

[19] A. W. Services, "What is synthetic data?" accessed: 08-10-2024. [Online]. Available: https://aws.amazon.com/what-is/synthetic-data/?nc1=h_ls

[20] H. Wu and F. No'e, "Variational approach for learning markov processes from time series data," *Journal of Nonlinear Science*, vol. 30, pp. 23 – 66, 2017.