Signal Prediction for Outlier Detection in Indoor Localization Using Transformers: A Multi-Dataset Benchmark

Ferrara Luigi
Department of Mathematics and Computer Science
Technische Universiteit Eindhoven
Eindhoven, Netherlands
l.ferrara@student.tue.nl

***Abstract*—This paper introduces a Transformer-based signal prediction module for an outlier detection system of WiFi RSSI fingerprinting in indoor localization. Our goal is to enable On-demand calibration by identifying signal outliers that indicate anomalous events, thus preventing costly manual recalibration.**

We developed an encoder-decoder architecture using a Convolutional Block and an Autoregressive Transformer to predict RSSI values by capturing long-range temporal dependencies via self-attention. Evaluations on the SODIndoorLoc and IPIN datasets show strong performance with MSE 0.0001 and NMSE 0.0036 for the first dataset, and MSE 0.0025 and NMSE 0.0564 for the other one, when training and testing are done within the same environment, confirming the model's ability to generalize in consistent settings. However, performance significantly degrades in cross-building scenarios due to environmental variations.

***Index Terms*—Indoor Localization Systems, Outlier Detection**

I. Introduction

Indoor localization has been widely studied in recent years. It is essential to detect device positions in bounded environments whenever other techniques, even widely known ones such as GPS [2], are not sufficient due to a lack of precision. Indoor localization systems can rely on different technologies. They can use Bluetooth [3], [9], geomagnetic signals [5], WiFi network [6], [7], [8] or other hybrid solutions [4]. One frequently used metric in WiFi-based indoor positioning is the Received Signal Strength Indicator (RSSI), which measures the power level of a received signal from a WiFi access point (AP). These values can be collected by mobile devices and used to estimate their position relative to known AP locations through fingerprinting (CITE) techniques. WiFi fingerprinting has been among the most studied methods for detection tasks and, especially when combined with deep learning techniques, has achieved promising results [10], [11]. This effectiveness is further supported by the availability of datasets based on WiFi RSSI values [17], [18], and [19], which have facilitated the study and advancement of this technology.

However, WiFi-based indoor positioning systems still face a serious limitation, that is, the need to periodically compensate radio map drifts when the network is affected by anomalous events [12]. In such cases, the detected device positions are biased because the radio map created from the network signals is shifted to a different frequency. An anomalous event is defined as any occurrence that can alter the indoor environment, such as a malicious non-cryptographic attack by adversaries, changes in the indoor setting, or unexpected issues affecting the wireless infrastructure (e.g., malfunction, misplacement, and removal of APs), including maintenance or outages [1].

Nevertheless, calibration has been at the core of various approaches to solving this problem over the years. In [1], additional sensors are deployed to capture dynamic changes in the radio map. However, this approach reduces the broad applicability of WiFi-based positioning and contradicts the goal of achieving ubiquitous computing. Crowd-sourced methods, such as those proposed in [13], [14], [15], and [16], reduce the need for manual recalibration by collecting user-contributed data from mobile devices. For instance, Redpin [13] offers an open-source Android application that allows users to voluntarily contribute fingerprints by tagging their current location on their smartphones. The application then collects signal strength data from nearby access points and transmits it to a server to update the radio map. Nonetheless, On-demand calibration [12] is preferable to reduce the maintenance costs of indoor localization systems. We define On-demand calibration as a process triggered when signal outliers are detected. If the number of outliers exceeds a certain threshold within a specific time window (possibly determined by a probabilistic model) a shift in the network can be inferred, indicating the occurrence of an anomalous event. To support this approach, it is crucial to identify outliers in the WiFi signal data accurately.

To this end, an outlier detection model is needed to successfully trigger On-demand calibration procedures and maintain a high level of reliability in an indoor localization system using WiFi fingerprinting. An effective outlier detector is a prerequisite for any On-demand calibration strategy, as it provides signal-level evidence needed to infer that an anomalous event has occurred and that the current radio map no longer reflects the true signal propagation in the environment. Once the model has been trained and can predict expected RSSI values accurately, it becomes relatively straightforward to flag singular measurements as outliers based on some distance metric (for instance k-nearest neighbors) by comparing observed values against the model's predictions.

Traditional outlier detection methods, such as statistical hypothesis tests, distance-based techniques (e.g., k-nearest neighbors [23]), and classical machine learning approaches such as SVM [24] or random forests [25], have the advantages of interpretability and low computational cost. However, since signal values are highly dependent on previous values of the same signal and because the distribution is rarely stationary in indoor environments, these models may fall short in capturing the true signal dynamics. Furthermore, many existing approaches in the indoor localization literature predate the introduction of Transformer architectures [20], and thus fail to leverage self-attention mechanisms which have demonstrated superior performance in modeling long-range dependencies, even when signals are used as input, as evidenced in biomedical applications [21] and in the context of speech recognition [22]. Because RSSI measurements rely on a long sequence of previous values, that is, long-range dependencies, self-attention is particularly well suited to capture these relationships. Hence, a Transformer-based model is the most sensible choice for effective outlier detection.

This study presents a novel contribution by thoroughly investigating a Transformer-based outlier detection system specifically built for RSSI fingerprinting data in indoor localization. To the best of our knowledge, no prior work has directly addressed the challenge of applying Transformer architectures to RSSI values for outlier detection.

This work aims to develop the first component of an outlier detection system capable of reliably identifying anomalies in WiFi signals, specifically the **signal prediction module**, with the goal of serving as a foundation for future anomaly detection frameworks in similar environments. With this study, we aim to address the following research questions:

Q1. Can the signal prediction module generalize across different WiFi signal datasets while maintaining reliable performance?

Q2. What insights can be drawn from evaluating the model across all available datasets?

This study is structured as follows: Section 2 reviews related work. Section 3 provides a detailed discussion of the datasets selected for evaluation. Section 4 presents the proposed encoder-decoder architecture, including the task formulation and model design. Section 5 reports experimental results and evaluates performance across datasets. Finally, Section 6 concludes with a summary of findings and directions for future research.

II. Related Works

A. Deep learning in Outlier Detection and Signal Processing

Several studies have addressed the challenge of detecting outliers in multivariate signal observations, where traditional univariate techniques often fall short due to increased dimensionality. Statistical approaches such as the Mahalanobis squared-distance (MSD) method model "normal" signals using a mean vector and covariance matrix, then flag new observations that deviate significantly. However, MSD's reliance on clean training data makes it susceptible to masking and swamping when outliers are already present in the dataset. To mitigate these issues, robust alternatives like the Minimum Volume Enclosing Ellipsoid (MVEE) and Minimum Covariance Determinant (MCD) have been introduced; by focusing on enclosing the bulk of the data or identifying a subset with minimal covariance, they improve resistance to contamination. Dimension-reduction techniques (most notably Principal Component Analysis) have also been employed, projecting high-dimensional signals into a lower-dimensional subspace where outlier-related variance can sometimes become more apparent. Yet, when normal observations themselves exhibit large variability, PCA can still suffer from masking effects, since the principal components may be dominated by benign variation rather than by anomalies. Regression-based methods offer an alternative by learning to predict the next time step of a signal; deviations between predicted and actual values can then indicate outliers. Such approaches are naturally suited to online monitoring, but they typically do not account for complex, high-dimensional correlations among multiple channels, limiting their effectiveness as dimensionality grows.

To overcome the limitations of purely statistical or linear techniques, researchers have turned to artificial neural networks (ANNs), leveraging their ability to approximate nonlinear relationships. Multilayer Perceptrons (MLPs) were among the first neural architectures applied to simultaneous feature learning and outlier classification, but they lack built-in mechanisms for capturing spatial or temporal structure. Convolutional Neural Networks (CNNs) extend MLPs by using convolutional filters to extract localized patterns in time-series data, and have been used to identify signals that deviate significantly from learned patterns. While CNN-based detectors can capture local anomalies, they often remain sensitive to adversarial perturbations or to benign but highly variable normal conditions, which can lead to false alarms. Recurrent Neural Networks (RNNs), especially those incorporating Long Short-Term Memory (LSTM) cells, have been employed to predict a statistical distance measure (e.g., an approximate Mahalanobis distance) at each time step; large residuals between predicted and observed distances flag potential outliers. Still, the sequential nature of RNNs makes them sensitive to the exact ordering of inputs, and their capacity to model very-high-dimensional dependencies is inherently limited by vanishing gradients and training complexity.

More recently, transformer architectures—originally developed for natural language processing—have shown considerable promise for various signal-based tasks, owing to their self-attention mechanism that captures global dependencies across all input positions. Unlike CNNs or RNNs, transformers can attend to every part of a multivariate signal simultaneously, allowing for the detection of anomalies that arise from complex, long-range interactions among channels. Positional embeddings enable transformers to encode temporal order without relying on strict recurrence, reducing sensitivity to input sequence shifts. Empirical evidence from computer vision has demonstrated transformers' robustness to occlusions, noise, and domain shifts (properties that translate well to outlier detection in high-dimensional signals). By leveraging attention scores, transformer-based models can potentially highlight exactly which time points or channels contribute most strongly to an anomalous observation, making them a promising candidate for advanced outlier detection.

B. Outlier Detection in WiFi Fingerprinting

In indoor WiFi localization systems, detecting anomalous RSSI values is essential to maintain an accurate radio map. One of the earliest dedicated systems is RAEDS by Zhang et al. [12], which combines a sliding-window autoregressive prediction filter with a multivariate nearest-neighbor predictor to flag individual outliers. The autoregressive filter assumes short-term RSSI stationarity and marks a reading as anomalous if its prediction residual exceeds a threshold derived from past errors. Simultaneously, the nearest-neighbor component identifies anomalies when the Euclidean distance between the current RSSI vector and its closest historical sample is unusually large.

While this dual-strategy approach was novel at the time, the AR filter is relatively simplistic and may struggle to capture complex signal dynamics in environments where multipath effects, interference, or device heterogeneity cause non-stationary behavior. Moreover, RAEDS and similar approaches have not been evaluated on multiple, heterogeneous WiFi datasets to assess generalization. In other words, it remains unclear whether a model calibrated on one building footprint or set of access points will continue to flag outliers accurately when deployed in a different environment or with a different AP layout. This gap is critical because, in practice, indoor environments can vary widely in terms of floor plan, device density, and AP placement.

In contrast, modern deep learning methods—particularly those based on self-attention—offer a more flexible framework for modeling RSSI sequences. By learning to attend to relevant past measurements across arbitrary time windows, a Transformer-based model can adapt more readily to signal fluctuations. Once such a model has been trained to predict expected RSSI values, it becomes straightforward to identify singular readings as outliers: one can compute a distance metric (e.g., Euclidean distance to the model's predicted vector or k-nearest neighbors in the embedding space) and flag measurements whose deviation exceeds a predetermined threshold. In this work, we address the limitations of RAEDS by (1) replacing the AR-NN hybrid with a Transformer-based predictor to improve prediction accuracy and (2) evaluating on multiple, publicly available RSSI datasets to demonstrate cross-dataset generalization.

III. Datasets Description

This study utilizes two publicly available datasets: the IPIN dataset [19] and the SODIndoorLoc dataset [26].

The SODIndoorLoc dataset is a large-scale WiFi-based indoor localization resource covering three buildings with around 8000 m^2, including corridors and rooms. It contains 23,925 samples from 1,802 unique locations, split into 21,205 training and 2,720 testing samples, with spatial resolution between 0.5 and 1.2 meters. Data were collected using 105 WiFi access points, and the dataset provides AP locations and CAD drawings. It supports tasks like classification, regression, clustering, and scene identification, making it suitable for evaluating anomaly detection and localization models. The IPIN dataset was collected over 325 grid points spaced 0.6 m apart, with sensor data including accelerometer and gyroscope readings, but this study focuses on WiFi RSSI values gathered via smartphone. Each point has RSSI readings from multiple APs, with missing detections set to -100 dBm. The dataset includes two campaigns, each with 325 samples and an average of 10.17 unique SSIDs per location, providing a solid basis for anomaly detection and localization evaluation.

IV. Design and Methodology

Our goal is to build an encoder-decoder architecture model capable of predicting Wi-Fi signals at a given time step, based on several complex channels. We begin by defining the decoding task and finally introduce the proposed end-to-end architecture in detail.

A. Task Formulation

Let us define a Wi-Fi signal over a time window as $X \in \mathbb{R}^{C\times T}$, where C is the number of channels and T is the number of time steps. These signals reflect the state of the Wi-Fi network over a given period. The decoding task consists of predicting the signal at time $t+1$, given a window of size W ending at time step t, that is, using the signals from time steps $t-W+1$ to t.

Thus, a supervised signal decoding task consists of finding a decoding function $f : \mathbb{R}^{C\times T} \to \mathbb{R}^{C}$. We denote by $Y = f(X)$ the predicted signal for a given input X.

B. Model Architecture

The first two components of the model are the Convolutional Block and the Transformer Encoder. They aim to find signal embeddings.

1) The Convolutional Block: The Convolutional Block is a convolutional neural network placed at the beginning of the model to begin processing the Wi-Fi signals. It enables the model to capture both local details and global dependencies from the input. The Convolutional Block transforms the input through two hidden layers:

$$H^{(1)} = \mathrm{ReLU}\big(\mathrm{BN}_1\big(\mathrm{Conv1}_{C\to 64}(X)\big)\big) \;\in\; \mathbb{R}^{64\times W},$$

$$H^{(2)} = \mathrm{ReLU}\big(\mathrm{BN}_2\big(\mathrm{Conv2}_{64\to 128}(H^{(1)})\big)\big) \;\in\; \mathbb{R}^{128\times W},$$

$$z = \mathrm{GAP}\big(H^{(2)}\big) \;=\; \frac{1}{W}\sum_{t=1}^{W} H^{(2)}_{:,t} \;\in\; \mathbb{R}^{128},$$

$$e = W_{\mathrm{fc}}\, z + b_{\mathrm{fc}} \;\in\; \mathbb{R}^{d}.$$

Where $\mathrm{Conv1}_{C\to 64}$ is a 1D convolution with kernel size 3 and padding 1, which maps C input channels to 64 output channels. The operator BN_1 denotes batch normalization applied on these 64 channels. Similarly, $\mathrm{Conv2}_{64\to 128}$ is a 1D convolution with kernel size 3 and padding 1 that maps 64 channels to 128 channels, followed by batch normalization BN_2 on the 128 channels. The symbol GAP stands for global average pooling performed along the temporal dimension. Finally, $W_{\mathrm{fc}} \in \mathbb{R}^{d\times 128}$ and $b_{\mathrm{fc}} \in \mathbb{R}^{d}$ represent the weight matrix and bias vector of the last fully-connected layer, respectively.

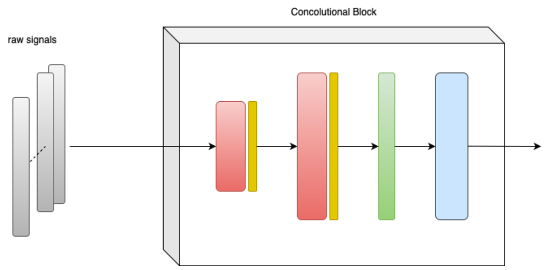


Fig. 1: A visual representation of the Convolutional Block. On the far left are the raw signals. The first and second convolutions are shown in red, followed by normalization layers in yellow. The green rectangle represents the global average pooling layer, and the last blue layer is an MLP that produces the final output.

The rest of the architecture of the signal predictor is made of an autoregressive transformer model.

2) The Autoregressive Transformer: The Autoregressive Transformer component processes the embeddings generated by the Convolutional Block to capture temporal dependencies and predict the next signal embedding in an autoregressive manner. It leverages the self-attention mechanism to weigh the importance of different past observations. The Autoregressive Transformer takes the sequence of d-dimensional embeddings e from the Convolutional Block as input. Let this input sequence be denoted as $E \in \mathbb{R}^{W'\times d}$, where W' is the window size (sequence length) and d is the embedding dimension. The transformation within the Autoregressive Transformer involves implementing the following:

$$p(x) = \prod_{i=1}^{n} p(x_i \mid x_1, x_2, \ldots, x_{i-2}, x_{i-1})$$

where $p(x)$ is the likelihood of the sequence x, and $p(x_i \mid x_1, x_2, \ldots, x_{i-2}, x_{i-1})$ is the probability of generating the i-th token given all previous ones. While this formulation is traditionally applied in natural language processing using word embeddings, in our case, we apply it to signal embeddings instead.

Positional Encoding: The input sequence E is first enhanced with positional information:

$$E_{\mathrm{pos}} = E + \mathrm{PE} \;\in\; \mathbb{R}^{W\times d},$$

where $\mathrm{PE} \in \mathbb{R}^{W\times d}$ represents the sinusoidal positional encodings. This step provides the model with knowledge of the sequence's element order. A dropout layer is then applied to E_{pos} for regularization.

Transformer Encoder: The positionally encoded sequence E_{pos} is then fed into a stack of Transformer Encoder layers. Each layer consists of a multi-head self-attention mechanism followed by a position-wise feed-forward network, with residual connections and layer normalization applied after each sub-layer. The output of the final encoder layer is denoted as $H_{N_{\mathrm{layers}}} \in \mathbb{R}^{W\times d'}$.

Output Layer: From the encoded sequence $H_{N_{\mathrm{layers}}}$, the embedding corresponding to the last time step is extracted. Let this be denoted as $h_{\mathrm{last}} \in \mathbb{R}^{d'}$. This h_{last} is then passed through a final fully-connected layer to produce the predicted next embedding $e_{\mathrm{pred}} \in \mathbb{R}^{d'}$:

$$e_{\mathrm{pred}} = W_{\mathrm{out}}\, h_{\mathrm{last}} + b_{\mathrm{out}} \;\in\; \mathbb{R}^{d'}.$$

Where $W_{\mathrm{out}} \in \mathbb{R}^{d'\times d'}$ and $b_{\mathrm{out}} \in \mathbb{R}^{d'}$ are the weight matrix and bias vector of the output fully-connected layer, respectively. This e_{pred} is the model's prediction for the embedding of the Wi-Fi signal at time $t+1$.

V. Evaluation

In this section, we evaluate the model on different datasets and buildings to assess its performance and ability

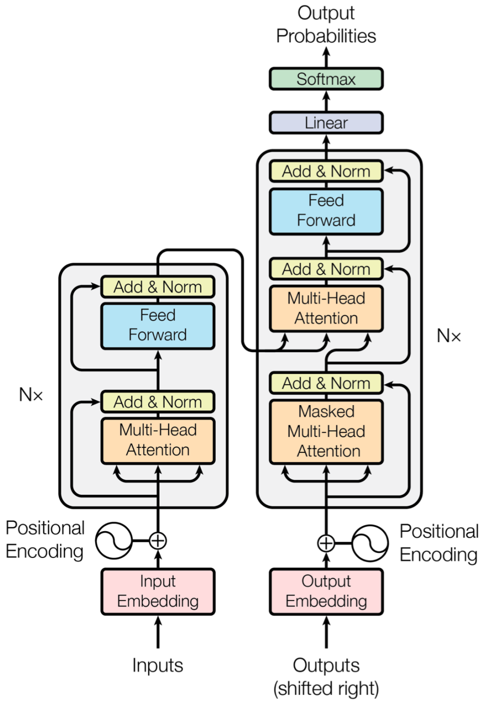


Fig. 2: Overview of the Transformer model architecture: the encoder is shown on the left and the decoder on the right.

to generalize across environments and data. The loss function used is the Mean Squared Error (MSE), which measures the average of the squared differences between predicted and actual values. Formally, it is defined as MSE $= \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$, where y_i denotes the true value, $\hat{y}_i$ the predicted value, and n the number of data points. A lower MSE indicates better model performance. The model is trained for 30 epochs in every experiment.

We present three evaluation scenarios. The first uses the SODIndoorLoc dataset [26], training and testing the model on signals produced within the same building. The second exploits the IPIN dataset [19], again training and testing on the same building. The third scenario uses the SODIndoorLoc dataset, but this time, training on data from one building and testing on another.

The results show that the model performs well when the training and testing data originate from the same environment. However, its performance degrades significantly when it is trained on one building and tested on another (see figure 5). This decline is likely due to variations in Wi-Fi signal characteristics, such as differences in the positions and types of access points between buildings, which hinder the model's ability to generalize effectively across environments. For additional experiments and detailed results, refer to the Appendix.

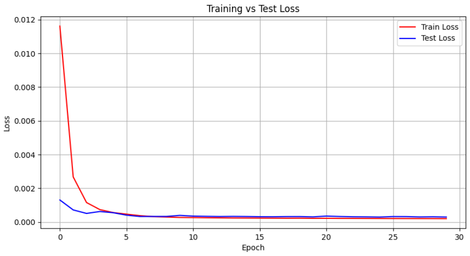


Fig. 3: Training loss (red) and test loss (blue) for the SYL building. The loss converges to zero after a few epochs, indicating good performance. The SYL building contains the most data points among all datasets considered, which helps the model generalize well. The model was trained for 30 epochs.

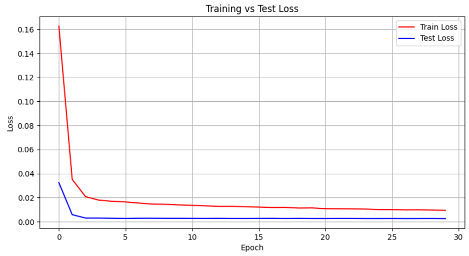


Fig. 4: Training loss (red) and test loss (blue) for the IPIN dataset. Although this dataset contains fewer samples than the previous one, the model still performs well. Training was conducted for 30 epochs.

To evaluate how well the model predicts the signal at the next time step, we compare the predicted signal with the true target value. For this comparison, we use one input sequence at a time and predict its next value. This process is repeated n times, where n is the size of the test dataset used in the experiment. For each prediction, we compute both the Mean Squared Error (MSE) and the Normalized Mean Squared Error (NMSE). The NMSE is calculated as the sum of squared errors between predicted and true values divided by the sum of the squared true values, i.e.,

$$\text{NMSE} = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} y_i^2},$$

where y_i is the true value and $\hat{y}_i$ is the predicted value. We use NMSE because it normalizes the error relative to the magnitude of the true signal, providing a scale-independent measure of prediction accuracy that is useful for comparing results across different datasets or signal ranges. A value of NMSE close to 0 indicates nearly perfect predictions, while a value equal to or greater than 1 suggests poor predictive performance.

In the SODIndoorLoc dataset, we evaluate the model on data from three different buildings: SYL, HCXY, and CETC331. Each with distinct environments, layouts, and access point deployments. This setup tests the model's ability to generalize across physical spaces. In the IPIN dataset, we use

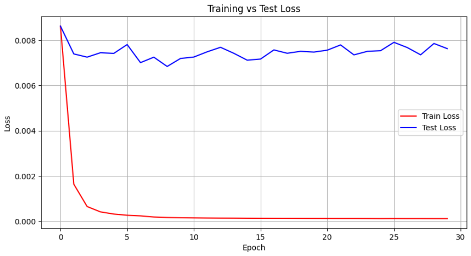


Fig. 5: Training loss (red) and test loss (blue) when training on the HCXY building and testing on CETC331. The model does not generalize well across buildings, as expected, due to differences in signal characteristics. The model was trained for 30 epochs.

data collected in the same environment but from two different devices, which we refer to as DEV1 and DEV2. This allows us to assess the model's robustness to device-related variability while keeping environmental conditions constant. The results for each source-target configuration are shown in the following tables:

TABLE I: Average MSE and NMSE across different source and target combinations

Source	Target	MSE	NMSE
SYL	SYL	0.0003	0.0066
HCXY	HCXY	**0.0001**	**0.0036**
CETC331	CETC331	0.0012	0.0206
SYL	HCXY	0.0016	0.0706
SYL	CETC331	0.0024	0.0425
HCXY	SYL	0.0034	0.0899
HCXY	CETC331	0.0076	0.1629
CETC331	SYL	0.0029	0.0719
CETC331	HCXY	0.0096	0.4419

TABLE II: Average MSE and NMSE for IPIN across devices

Source	Target	MSE	NMSE
DEV1	DEV1	0.0030	0.0608
DEV0	DEV0	**0.0025**	**0.0564**

As shown in Table I, the best predictive performance occurs when training and testing on the same building, with the lowest MSE and NMSE values. Notably, the HCXY building achieves the best results among the three buildings. This is likely because HCXY contains the largest number of data samples, allowing the transformer model to learn more comprehensive patterns and thus generalize better within that environment.

Cross-building predictions result in higher errors, reflecting the difficulty of generalizing across different physical layouts and access point configurations.

Table II shows the metrics for the IPIN dataset. Predictions on the same device (DEV0 or DEV1) demonstrate low error.

Overall, the model performs best when training and testing data come from the same building or device, with performance strongly influenced by the quantity and quality of available training data.

VI. Conclusions and Future Works

The results demonstrate that the proposed model performs well when trained and tested on data collected from the same building. This is reflected in the low Mean Squared Error (MSE) and Normalized MSE (NMSE), indicating that the model is capable of accurately predicting WiFi signal patterns within a consistent environment. This finding holds across both the SODIndoorLoc and IPIN datasets, despite their differences in hardware, acquisition protocols, channels, and physical layouts.

However, when the model is trained on data from one building and tested on data from a different building, its performance degrades significantly. The test MSE fluctuates and reveals the model's limited ability to generalize across environments. This is most probably because WiFi signals are highly sensitive to spatial configuration, the positions of access points, and the surrounding physical context, all of which differ substantially between buildings. These results suggest that when trained on representative and diverse data, the model is capable of generalizing effectively within the same environment, addressing our first research question (**Q1**).

Evaluating the model across both datasets leads to several insights. First, it confirms that WiFi fingerprinting models are highly sensitive to environmental characteristics. Second, it shows that despite the heterogeneity of the datasets, deep learning models can still learn useful signal representations, provided the training and testing conditions are aligned. Third, it highlights the limitations of current models in cross-domain scenarios, pointing to a need for domain adaptation techniques or more robust architectures if transfer across buildings is required (**Q2**).

This study introduces a promising Transformer-based approach for outlier detection in RSSI fingerprinting, with future work focusing on integrating the model into an automated calibration system that updates radio maps based on detected anomalies. Challenges remain regarding real-time deployment due to computational demands, suggesting the need for model optimization. Additionally, exploring Vision Transformers (ViTs) [28] by converting RSSI data into visual formats like heatmaps could enhance anomaly detection by leveraging spatial correlations. Comparing the current Transformer model with a ViT-based approach may reveal which method better captures subtle signal anomalies [27].

References

[1] Y. Chen, J. Chiang, H. Chu, P. Huang, A. Tsui, "Sensor-assisted wi-fi indoor location system for adapting to environmental dynamics" Association for Computing Machinery, New York, NY, United States, 10 October 2005.

[2] Xu, Guochang, and Yan Xu. GPS. Vol. 2. Springer-Verlag Berlin Heidelberg, 2007.

[3] Bruno, R., Delmastro, F. (2003). Design and Analysis of a Bluetooth-Based Indoor Localization System. Personal Wireless Communications. PWC 2003. Lecture Notes in Computer Science, vol 2775. Springer, Berlin, Heidelberg.

[4] Baniukevic, Artur, et al. ”Improving wi-fi based indoor positioning using bluetooth add-ons.” 2011 IEEE 12th International Conference on Mobile Data Management. Vol. 1. IEEE, 2011.

[5] Al-Homayani, Fahad, and Mohammad Mahoor. ”Improved indoor geomagnetic field fingerprinting for smartwatch localization using deep learning.” 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN). IEEE, 2018.

[6] Song, Xudong, et al. ”Cnnloc: Deep-learning based indoor localization with wifi fingerprinting.” 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI). IEEE, 2019.

[7] Abbas, Moustafa, et al. ”WiDeep: WiFi-based accurate and robust indoor localization system using deep learning.” 2019 IEEE International Conference on Pervasive Computing and Communications (PerCom. IEEE, 2019.

[8] Ravi, Anuradha, and Archan Misra. ”Robust, fine-grained occupancy estimation via combined camera & WiFi indoor localization.” 2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS). IEEE, 2020

[9] Altini, Marco, et al. ”Bluetooth indoor localization with multiple neural networks.” IEEE 5th International Symposium on Wireless Pervasive Computing 2010. IEEE, 2010.

[10] Brattinga, Martijn. LSTM-based Indoor Localization with Transfer Learning. BS thesis. University of Twente, 2022.

[11] Luo, Xuanshu, and Nirvana Meratnia. ”A geometric deep learning framework for accurate indoor localization.” 2022 IEEE 12th International Conference on Indoor Positioning and Indoor Navigation (IPIN). IEEE, 2022.

[12] Zhang, Dezhi, et al. ”Crowdsourcing based radio map anomalous event detection system for calibration-on-demand.” 2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN). IEEE, 2014.

[13] Bolliger, Philipp. ”Redpin-adaptive, zero-configuration indoor localization through user collaboration.” Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments. 2008

[14] Ledlie, Jonathan, et al. ”Molé: A scalable, user-generated WiFi positioning engine.” Journal of Location Based Services 6.2 (2012): 55-80.

[15] Rai, Anshul, et al. ”Zee: Zero-effort crowdsourcing for indoor localization.” Proceedings of the 18th annual international conference on Mobile computing and networking. 2012.

[16] Radu, Valentin, and Mahesh K. Marina. ”HiMLoc: Indoor smartphone localization via activity aware pedestrian dead reckoning with selective crowdsourced WiFi fingerprinting.” International conference on indoor positioning and indoor navigation. IEEE, 2013

[17] Klus, Lucie, et al. ”TUJI1 Dataset: Multi-device dataset for indoor localization with high measurement density.” Data in Brief 54 (2024): 110356.

[18] Bi, Jingxue, et al. ”Supplementary open dataset for WiFi indoor localization based on received signal strength.” Satellite Navigation 3.1 (2022): 25.

[19] Barsocchi, Paolo, et al. ”A multisource and multivariate dataset for indoor localization methods based on WLAN and geo-magnetic field fingerprinting.” 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN). IEEE, 2016

[20] Vaswani, Ashish, et al. ”Attention is all you need.” Advances in neural information processing systems 30 (2017).

[21] Lee, Young-Eun, and Seo-Hyun Lee. ”EEG-transformer: Self-attention from transformer architecture for decoding EEG of imagined speech.” 2022 10th International winter conference on brain-computer interface (BCI). IEEE, 2022.

[22] Radford, Alec, et al. ”Robust speech recognition via large-scale weak supervision.” International conference on machine learning. PMLR, 2023.

[23] Cover, Thomas, and Peter Hart. ”Nearest neighbor pattern classification.” IEEE transactions on information theory 13.1 (1967): 21-27.

[24] Vapnik, Vladimir, Steven Golowich, and Alex Smola. ”Support vector method for function approximation, regression estimation and signal processing.” Advances in neural information processing systems 9 (1996).

[25] Ho, Tin Kam. ”Random decision forests.” Proceedings of 3rd international conference on document analysis and recognition. Vol. 1. IEEE, 1995.

[26] Bi, J., Wang, Y., Yu, B. et al. (2022), “Supplementary open dataset for WiFi indoor localization based on received signal strength”, Satellite Navigation, Vol. 3, No. 25 (2022)

[27] Rafique, Hamaad, et al. ”Fusing Visuals with Magnetic Signals to Improve Indoor Localization Using Vision Transformer.” 2024 14th International Conference on Indoor Positioning and Indoor Navigation (IPIN). IEEE, 2024.

[28] Dosovitskiy, Alexey, et al. ”An image is worth 16x16 words: Transformers for image recognition at scale.” arXiv preprint arXiv:2010.11929 (2020).

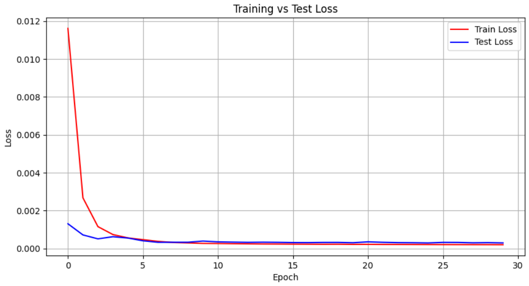
Planning

Week(s)	Subject
3 & 4	Drafting initial paper content
5	Finalizing *Introduction* and *Related Work*
6	Finilazing *Introduction* and *Related Work* and start build the transformer
7	Continue working on code
8	still code + design and methodology
9	design and methodology + all the missing sections + presentation

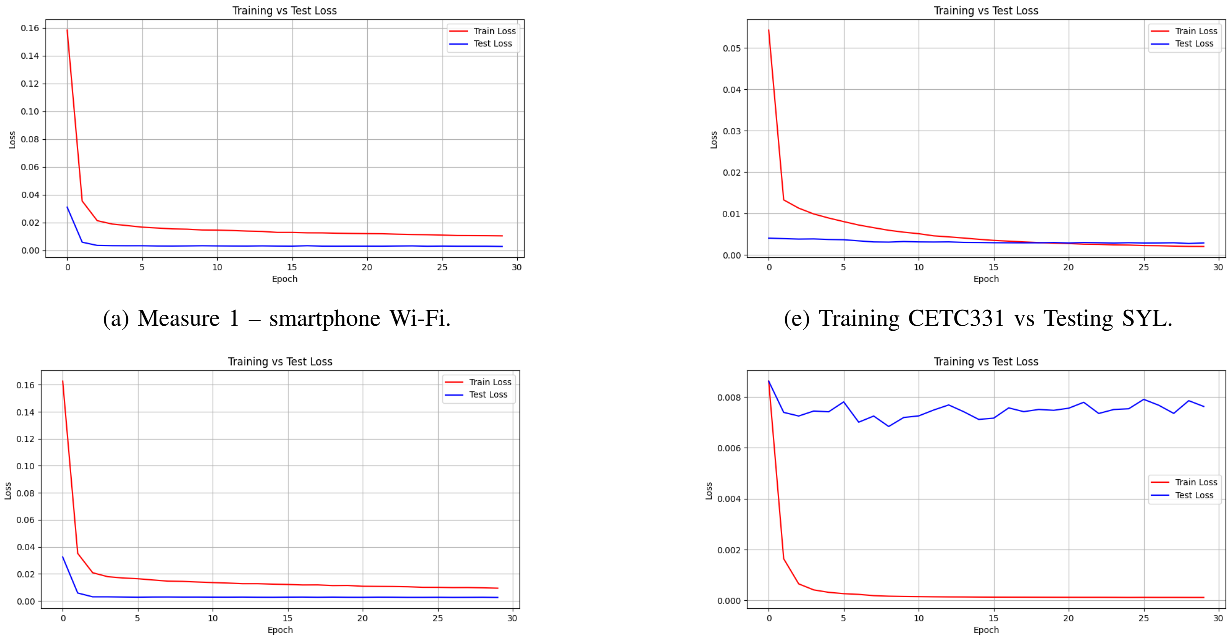
Reflection

have participated actively and attentively in most lectures and quizzes. While writing the draft paper, I found the course readings very informative. In addition, I conducted external research to consult multiple sources and broaden my understanding, even though some papers occasionally led to conceptual blind spots. I do not yet have results from my own research, but I expect them to offer valuable insights into best practices for working with RSSI fingerprinting, particularly regarding preprocessing. I chose this topic because data representations and transformer models can produce unexpected results, and I wanted to explore these aspects more deeply

Appendix



(a) Training vs Testing on SYL.

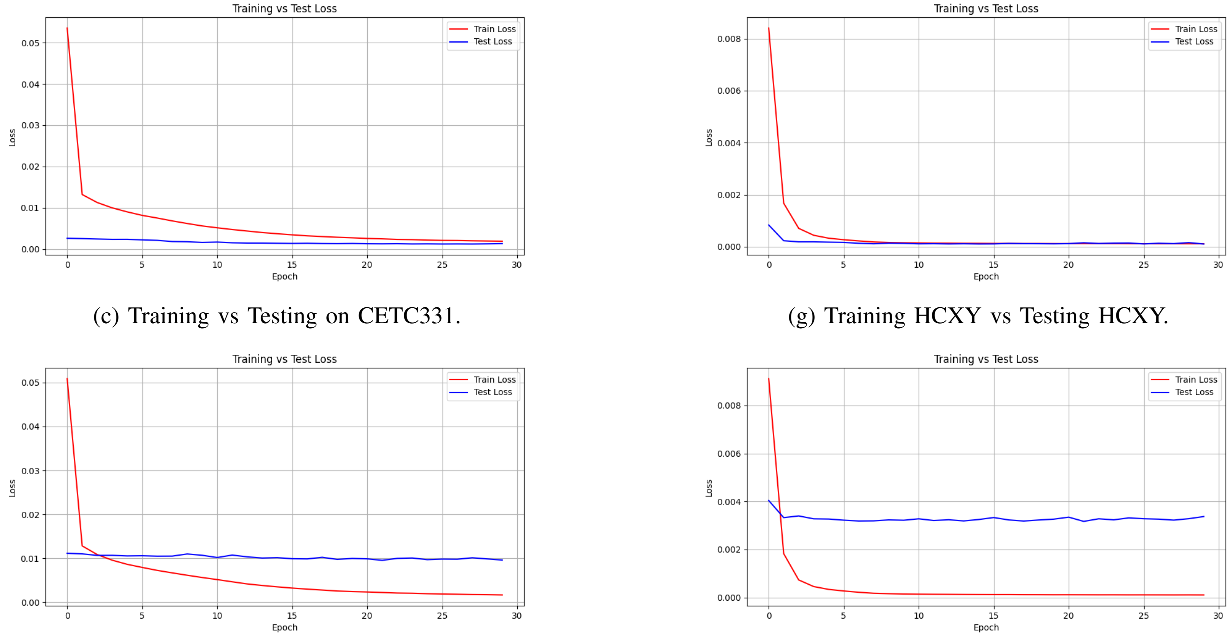


(a) Measure 1 – smartphone Wi-Fi.

(b) Measure 2 – smartphone Wi-Fi.

(e) Training CETC331 vs Testing SYL.

(f) Training HCXY vs Testing CETC331.

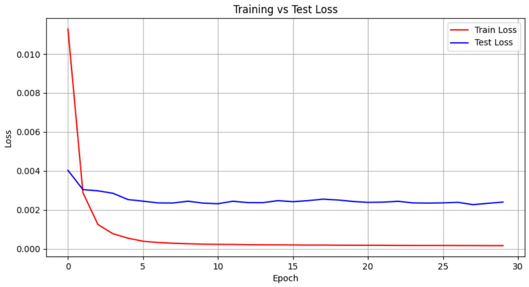


(c) Training vs Testing on CETC331.

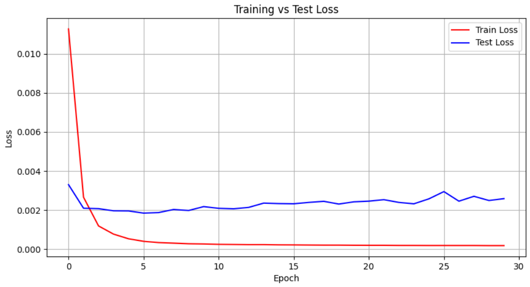
(d) Training CETC331 vs Testing HCXY.

(g) Training HCXY vs Testing HCXY.

(h) Training HCXY vs Testing SYL.



(i) Training SYL vs Testing CETC331.



(j) Training SYL vs Testing HCXY.