WiFi Based Distance Estimation Using Supervised Machine Learning

Kahraman Kostas†, Rabia Yasa Kostas‡, Francisco Zampella*, Firas Alsehly*

*Edinburgh Research Center, Huawei Technologies Co., Ltd. United Kingdom
†School of Mathematical & Computer Sciences, Heriot Watt University, Edinburgh, United Kingdom
‡School of Informatics, University of Edinburgh, Edinburgh, United Kingdom

{kahraman.kostas, rabia.yasa.kostas, francisco.zampella, firas.alsehly}@huawei.com

Abstract—In recent years WiFi became the primary source of information to locate a person or device indoors. Collecting RSSI values as reference measurements with known positions, known as WiFi fingerprinting, is commonly used in various positioning methods and algorithms that appear in literature. However, measuring the spatial distance between given set of WiFi fingerprints is heavily affected by the selection of the signal distance function used to model signal space as geospatial distance. In this study, the authors proposed utilization of machine learning to improve the estimation of geospatial distance between fingerprints. This research examined data collected from 13 different open datasets to provide a broad representation aiming for general model that can be used in any indoor environment. The proposed novel approach extracted data features by examining a set of commonly used signal distance metrics via feature selection process that includes feature analysis and genetic algorithm. To demonstrate that the output of this research is venue independent, all models were tested on datasets previously excluded during the training and validation phase. Finally, various machine learning algorithms were compared using wide variety of evaluation metrics including ability to scale out the test bed to real world unsolicited datasets.

Index Terms—WiFi fingerprinting, RSSI, machine learning, distance estimation indoor positioning, supervised learning

I. INTRODUCTION

In the last decade, the proliferation of smart devices has increased significantly, their processing power and capabilities allow users to access services that facilitate their life like communication, wireless payment, messaging, navigation, maps, etc. Many of those services rely on the user’s position to adjust the information shown, and the most common way to obtain it outdoor, is through Global Navigation Satellite Systems (GNSS) like GPS, GLONASS, BEIDOU or GALILEO. In indoor environments like offices, shopping malls, subway stations, among others, the common no line of sight to the satellites prevent it from providing an accurate position [1] and the smart devices require additional sources to obtain the location information.

Many technologies had been used to estimate the position of a user indoors [2], including ultrasound [3], visible light communication [4], visual odometry [5], magnetic fields [6], dead reckoning [7], Bluetooth [8], Ultra Wide Band [9], Wi-Fi [10], among others. This research will focus in the use of Wi-Fi technology and its use in positioning due to its wide availability in all of those indoor environments. A Wireless Access Point (WAP) can provide different measurements to use for positioning, however the Received Signal Strength (RSS) [11] has become the most common information source due to its low cost and easy implementation [12]. This measurement can be used in many different ways to estimate positions.

The most common approach to address the non regularity of the power decay with respect to the position is to take samples (fingerprints) of the RSS values from multiple WAPs in a grid of reference points. The position of a new fingerprint can then be estimated from the similarity or distance between the new signal and the previous reference points [14]. The signal differences can use several metrics (Euclidean, Manhattan, Jaccard, Cosine, etc. [15]), and from those values, the position can be estimated using deterministic, probabilistic or pattern recognition methods.

It has been observed that the signal difference used affects significantly the performance of the positioning algorithm and the authors in this paper propose to use data science methodologies to improve the estimation of the spatial distance between 2 fingerprints. Most fingerprint and distance estimation methods are based on specific datasets of almost laboratory conditions, the authors propose to use a multi-environment training, and isolated dataset testing as a prediction of the performance of the distance estimation in unknown environments.

The remainder of this paper is structured as follows: in Section II related studies in the field are discussed, in Section IV contribution and how the method serve the Indoor Positioning System (IPS) are described, in Section V the system architecture and details of the experiments are explained. Section VII contains the results of the experiments and discussion, and in Section VIII the conclusions are presented.

II. RELATED WORK

Recently, many Artificial Intelligence (AI) techniques; such as Machine Learning (ML) and Deep learning (DL) have
been used in the solution of indoor positioning problems. The ability of data science to solve high complexity real life problems without the need for exact rules and equations has made AI methods very attractive [16], [17] for positioning problems. In the IPS field, RADAR [18] was one of the first research works to propose a ML based solution using K-nearest neighbours (KNN) to estimate the location of a user with a few meters error distance. Since then other algorithms were also tested; Gradient Boosted Classifier (GBC) and Random Forest Classifier (RF) in [19], Deep Neural Networks (DNN) based classification in [20], and novelty Graph-based classification algorithm (Hierarchical Navigable Small World Graphs) in [21].

KNN and its extension weighted K-nearest neighbours (WKNN) have remained quite common for indoor positioning using a reference set, and it is usually referred as fingerprinting. The work of [18] was based on the Euclidean distance between the fingerprints, but as many authors have highlighted, the accuracy of the system is heavily affected by the distance metric used [15], [18], [22]. That distance estimation can be applicable not only for localization, but also tracking of mobile devices and its interactions. This can be valuable for the inference of the network of social relationships, and recently it has helped in the contact tracing of positive cases of COVID-19. In [23] and [19], the authors deduced the proximity network, based on the trajectories and WiFi signal distances between mobile devices.

Currently, the most common distance (or feature) used for fingerprinting positioning is the Euclidean distance, but subsequent studies improve the metric by extending the feature list with new distances; Jaccard, Manhattan [19], Kendall [20], correlation coefficients [24], [25], Sørensen and 53 other distances [15]. Although, most works measure the quality with one distance or a combination of a few at a time, only [20] tries to estimate the distance using many distance features, training a deep neural network to estimate the real distance from 14 distance features.

These applications are trained and tested on the data obtained from single buildings such as universities [26], [27] or the well known UJI Indoor Localization dataset [15], [21] that provides real life data from multiple floors.

III. Motivation

Today, people spend a lot of time in large buildings such as hospitals, airports, shopping malls. It is very important, sometimes even vital, to find directions or locate personnel and equipment in a building. In such areas, where satellite systems do not work efficiently, positioning using WiFi signals and machine learning methods are quite common. However, most of the common studies train machine learning models using data collected only from a single or few collocated buildings. Furthermore, it is widely common to use the same venue for training and testing which render such approach as prone to overfitting and unreliable in unknown environments. To deal with this problem, we propose a generalized model trained as venue independent. We expect that this model will have a wider use with a better extrapolation to unknown scenarios. Furthermore, It has been observed that most available research in this concept use a single or limited combination of signal modelling distance functions to estimate the similarity between 2 fingerprints. Given that each distance function usual provide context for a specific characteristics in the signal distribution, which obviously cannot work everywhere, we propose to analyse the influence of each function separately as a feature in our distance modelling. Therefore, expanding the context of one machine learning model to utilize more attributes and functions of available WiFi data has motivated our work in this paper.

IV. Contribution

In this study, we proposed a distance estimation model that detects whether two fingerprints are close to each other and estimates the approximate distance between RSS fingerprints using supervised ML methods; regression and classification. With this study, we have made the following contributions.

• We conducted a research with focusing on reproducibility. Our steps are easy to follow to create a new models and applicable on new data from different structures; buildings, train stations, airports etc.
• We investigate a broad spectrum of ML and DL algorithms. We assume that it will be a good reference point to comparison for future studies.
• Instead of using one dataset obtained from one building, which used frequently for both training and testing purposes in previous studies, we have reserved 2 datasets for testing purposes only, 13 different datasets in total. We used strictly isolated training and testing data collected from separate buildings. In order to optimize the parameters and estimate distances initially, we used third dataset named validation. In this way, we think that more generalizable models and realistic results can be obtained.
• We do not only estimate the distance between two points (with regression), but also measure the success of our system in predicting the closeness of results (with classifier).
• We made the dataset, code used and results obtained publicly available1.

V. Methodology

In this section, dataset, feature extraction, features, ML algorithms and evaluation metrics are explained.

A. Data

Many studies uses datasets from the same building for the model training and testing. However, the goal of this paper is to create a generalizable model that can be applicable in any indoor environments. In this respect, while making the dataset that will be used in our study, it was prioritized to have a high representation ability. For this purpose, 11 datasets

1Source code available at: github.com/kahramankostas/WiFi-Fingerprint
were used [28]–[38] from the IPIN conference resources and additionally two datasets collected by Huawei. The most important difference of Huawei datasets from other datasets is that the data is collected from shopping centers. Therefore, the new datasets better represent the indoor data samples encountered on daily basis.

![Diagram](image)

Fig. 1: The creating process of training, validation, testing and isolated test datasets from the data.

A large dataset was created by combining the 10 IPIN datasets [28]–[37] and one of the Huawei datasets [39]. Thus, we ensured that it contains data from different building structures. The dataset was partitioned into three sets (70%, 15%, and 15%) for training, validation and testing purposes respectively. The validation data was used in the evaluation process of every stages, while retaining the test data for final evaluation independent and isolated. On the other hand, one of the Huawei datasets [40] and one of the IPIN datasets [38] were isolated from the initial training, testing and validation data to evaluate the generalization of the system. The creation process of the datasets is also visualized in Fig. 1.

### B. Feature Extraction

The distribution of the Received Signal Strength Indicator (RSSI) values in the datasets is given in Fig. 2. Almost all data are tucked between -20 decibel (dB) and -95 dB, except for some exceptional values in this figure. Ignoring RSSI values less than -95 dB and greater than -20 dB, outliers are cleared. In the feature extraction process, RSSI values of the fingerprints were converted into vectors. The distance between these vector pairs which representing fingerprints, was calculated with various distance-similarity calculation methods. The names and formulas of these calculation methods are given in Table I. However, the fingerprints in the dataset are quite different from each other in terms of both the size and the device they contain. Fig. 3 shows the distribution of fingerprints according to the number of devices they have in Huawei datasets. While calculating the features, fingerprints intersected 2 or more Medium Access Control (MAC) addresses were taken into account, the others were excluded. In addition, only RSSI values of intersecting MAC addresses are taken into account while calculating the distance, and values of non-intersecting MAC addresses are ignored. Since datasets do not share any MAC addresses with each other, the distance between fingerprints in different datasets is not calculated. Two more features have been added that represent the number of MAC addresses contained by fingerprint pairs. These are the number of intersecting and combination MAC addresses. The actual distance between fingerprints pairs is used as the label value. The distribution of the distance between the fingerprints in the datasets is given in Fig. 4.

![Diagram](image)

Fig. 2: The distribution of RSSI values in all datasets.

| Name         | Formula                                                                 |
|--------------|--------------------------------------------------------------------------|
| Bray-Curtis  | \( \sum_{i} |w_i - v_i| \)                                                                 |
| Canberra     | \( \sum_{i} |u_i - v_i| \)                                                                 |
| Chebyshev    | \( \max_i |u_i - v_i| \)                                                                 |
| City Block   | \( \sum_i |u_i - v_i| \)                                                                 |
| Correlation  | \( 1 - \frac{\langle u - \bar{v}, v - \bar{u} \rangle}{\|u - \bar{v}\| \|v - \bar{u}\|} \) |
| Cosine       | \( 1 - \frac{\bar{u} \cdot \bar{v}}{\|u\| \|v\|} \)                      |
| Euclidean    | \( \sqrt{\sum_{i=1}^{p} (u_i - v_i)^2} \)                                 |
| Jaccard      | \( \frac{|u_i \cap v_i|}{|u_i + v_i - u_i \cap v_i|} \)                   |
| Jensen-Shannon| \( \sqrt{\frac{D(u|m|) + D(v|m|)}{2}} \)                               |
| Minkowski    | \( \sum_{i=1}^{p} |u_i - v_i|^p \)                                      |
| Squared Euclidean| \( \sum_{i=1}^{p} (u_i - v_i)^2 \)                                    |
| Weighted Minkowski| \( (\sum_{i=1}^{p} |u_i - v_i|^p) \right)^{1/p} \)                        |

2These datasets were presented to the competitors in the University Challenge Competition 2021 with the concept of "Data Science for Indoor positioning" organized by Huawei-UK.
We used all the features and selected features are listed. Hyperparameter Optimisation was used.

### C. Evaluation Metrics

To evaluate the results, the Root mean squared error (RMSE) was used as:\[
\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}
\]
(y represents the actual value, \(\hat{y}\) the estimated value and, \(n\) number of samples).

In addition to RMSE, the problem was translated into a classification, using a limit value (4 meters in this case). The value below this limit is accepted as TRUE and the value above it as FALSE, and the regression problem is transformed into binary classification. Using this method, the recall can be calculated as:\[
\text{Recall} = \frac{\text{True Positive (TP)}}{\text{TP} + \text{False Negative (FN)}}
\]
and the precision as:\[
\text{Precision} = \frac{\text{True Positive (TP)}}{\text{TP} + \text{False Positive (FP)}}
\]
Due to level of noise in the data, our scoring metric for is biased towards precision over recall. To give a higher weight to the precision the \(F_\beta\) score was used (with a \(\beta\) value of 0.05), calculated as:

\[
F_\beta = \frac{(1 + \beta^2) \times \text{Precision} \times \text{Recall}}{\beta^2 \times \text{Recall} + \text{Precision}}
\]

### D. Algorithm Selection

In the datasets, the actual values of the desired outputs along with the fingerprints are shared as the distance in meters between the test points. Based on these data, we can say that this problem is a supervised learning and regression problem.

| Machine Learning Type | Machine Learning Algorithm |
|-----------------------|-----------------------------|
| Classical linear regressors | Linear Regression (LR) |
| Tree based | Decision Tree (DTR) |
| Bayesian method based | Bayesian Ridge (BR) |
| Kernel method based | Linear Support Vector (LSVR) |
| Ensemble methods based | eXtreme Gradient Boosting (XGBR) |
| Instance based | K-Nearest Neighbors (KNNR) |
| Deep Learning | Artificial Neural Networks (ANN) |

### VI. Experiments

Experiments section consists of four subsections. In the first part, an initial evaluation is made using different ML methods with training and validation datasets. In the second part, ideal limits are found by filtering the training data with different thresholds. In the third section, feature selection is made using feature importance scores and genetic algorithm (GA). After that, the ideal hyperparameters are found for each ML algorithm by optimization. In the last part, all ML models are tested with isolated datasets in order to measure the generalizability of the obtained methods. The experiment process is visualized in Fig. 5.

![Fig. 3: The distribution of fingerprints according to the number of MAC addresses they contain (Huawei datasets [39], [40])](image3)

![Fig. 4: The distribution of distances between fingerprint pairs.](image4)

![Fig. 5: Steps of the experimental process.](image5)

**A. Choosing Machine Learning Method**

As an exploratory step, we performed initial experiments using 14 features and the ML methods given in Table II. Table III shows the results obtained from this process. Among ML methods XGB has the highest score followed by ANN.
and KNN very closely. However, KNN is particularly notable for its extremely high inference time. In real life solutions, it can be a disadvantage. Although LR and NB are two methods with the lowest scores, they stand out with low training and inference time.

In the next steps, we will observe the results of the LR algorithm in the stages of reshaping our data and feature selecting. The main reason for selecting LR needing less training and inference time in these time-consuming processes. In addition, because it is very simple and deterministic algorithm it does not contain multiple variables e.g., hyper-parameters, random seed and gives stable results.

### TABLE III: Comparison of ML algorithms with average of 10 repeats using validation dataset. t is time in seconds. Standard deviation is given with $F_β$. The best values are underlined. Hardware: Intel-Core i7-8565U 1.99GHz CPU, 16GB RAM.

| ML | Precision | Recall | $F_β$ | RMSE Train-t | Test-t |
|----|-----------|--------|-------|---------------|--------|
| DTR | 0.368 | 0.374 | 0.368±0.000 | 23.898 | 1294.1 |
| LR | 0.168 | 0.031 | 0.166±0.000 | 23.345 | 17.798 |
| BR | 0.168 | 0.031 | 0.166±0.000 | 23.345 | 10.646 |
| KNNR | 0.626 | 0.215 | 0.623±0.000 | 21.118 | 637.70 |
| XGBR | 0.667 | 0.160 | 0.662±0.000 | 19.271 | 341.44 |
| LSVR | 0.209 | 0.308 | 0.206±0.193 | 110.38 | 17647 |
| ANN | 0.631 | 0.359 | 0.624±0.065 | 19.961 | 6434.8 |

### B. Filtering Training Data

We have shared the distribution of the distances in Fig. 4. According to this figure, the distance between fingerprints varies between 0 and 160 meters and it is quite scattered. In most cases, as the distance between two fingerprints increases, the RSSI decreases and correspondingly, the noise level in the dataset will increase. This noise is likely to affect the model negatively. Therefore, we observed the change of the model by adding various thresholds at training data. For example, a threshold value of 20 means that only fingerprint pairs that are 20 meters or less apart are used in training. In Fig. 6 the $F_β$ results with different thresholds are shared. According to this plot $F_β$ increases between 15 and 25 metres, it starts to decrease after 25. With setting the threshold to 25, we increased the $F_β$ score from 0.17 to 0.52 for LR. So, we will use the training dataset limited 25 metres which is the peak value, in our next step.

### C. Feature Selection

The model is only as good as the features it uses. That’s why we’re going to select the most efficient ones among 14 features. In the first stage of this process, we will use a voting system (pypi.org/project/xverse/) to see the importance level of each features. This voting system uses six different feature importance scoring methods (chi-square, extra trees, information value, L1-based, recursive feature elimination, and random forest). Each of these methods votes for a feature as useful or not. The features and the votes they received from each method are given in Fig. 7. Although the feature scores give a general idea about the features, they do not allow us to understand which feature group is better. To make up for this point, we used an wrapper method, the GA. It takes $2^{14}$ trials to discover the best feature set of 14 features by trial and error. The GA can create feature sets that perform very well with reasonable computational cost. The $F_β$ change occurring parallel to the generations in GA can be seen from the Fig. 8. When all features were used, the score was around 0.53. After GA applied it increased to over 0.65. The GA discarded four out of 14 features and introduced a new feature set with 10 features. When we look at the eliminated features importance scores, three of them (S euclidean, Cityblock, and Uniting MAC) can be classified as unimportant, while the other feature (Braycurtis) is classified as quite important. An insignificant feature can become more important if it is combined with another feature. On the other hand, an important feature may undermine the model by conflicting with another feature or features. Braycurtis feature may have disabled because it conflicts with other features.

After finding the ideal feature set with feature selection, we performed hyper-parameter optimization for the MLs. We used the random search method (as used by scikit-learn.org/stable/) for optimization. So, we were able to search for a wide range of parameters within reasonable computational time. In all our processes so far, we have only used the validation dataset to evaluate the performance of steps.

### D. Final Experiments

So far, we have carried out all experiments and studies using LR. In this step, we adapted all of the experiments to other ML algorithms and reported our final results. For all ML methods, we created models using the training set filtered with a distance of 25 meters and 10 selected features and measured their performance level. In this measurement, in addition to the validation we used at the beginning, we also used other datasets that we isolated. For this, the model created by using the training dataset was tested on four datasets. After the model training was done once, the same model was used in all four datasets, and the operations were repeated 10 times. The results are given in Table IV.

3selected features are: Correlation, Chebyshev, Intersecting MAC, Euclidean, Cosine, Jensenshannon, Jaccard, Canberra, Minkowski, and Wminkowski
features

Fig. 7: Features and their votes they receive from feature importance methods.

|                | ML      | Precision | Recall     | \( F_\beta \) | RMSE   | Train-t | Test-t |
|----------------|---------|-----------|------------|---------------|--------|---------|--------|
| DTR            | 0.796   | 0.042     | 0.759±0.023| 35.847        | 27.605 | 1.192   |
| LR             | 0.809   | 0.010     | 0.675±0.000| 36.180        | 3.132  | 0.173   |
| BR             | 0.809   | 0.010     | 0.675±0.000| 36.180        | 4.512  | 0.157   |
| KNNR           | 0.606   | 0.104     | 0.599±0.000| 35.906        | 139.254| 398.1   |
| XGBR           | 0.791   | 0.065     | 0.769±0.000| 36.191        | 1969.59| 10.54   |
| LSVR           | 0.373   | 0.054     | 0.340±0.244| 35.453        | 2062.70| 0.142   |
| ANN            | 0.812   | 0.032     | 0.754±0.042| 35.730        | 4038.79| 114.2   |

TABLE IV: Comparison of ML algorithms with average of 10 repeats using four datasets. t is time in seconds. Standard deviation is given with \( F_\beta \). The best values are underlined.

Fig. 8: The change in the \( F_\beta \) with generations in the GA.

VII. RESULTS AND DISCUSSION

When we compare the validation part of our final results with the initial experiments, there is a high increase in the success of each of the seven ML algorithms. When we take an overview of Table IV, the XGBR method is the most successful method for all datasets, results varying between 0.76 and 0.90, in terms of \( F_\beta \) score. It is followed by DT, KNN and ANN methods. If we focus on methods in terms of speed, the XGBR method is reasonably fast, besides its outstanding success. However, the extremely fast and reasonable results of DT may make this method a suitable choice in real life applications. Although LR and BR, which have very short inference times, achieved remarkable success in test and validation datasets, they could not maintain their performance on isolated datasets. The reason for the lower success may be that LR (and similar method BR) is biased against validation and test sets. Since the validation dataset has been used many times with LR in the experimental steps such as feature selection and filtering, this caused an overfitting against the validation and test set (because of both them are created with the training dataset, they have a similar structure). However, the experimental steps contributed to all other ML methods as well.

The proposed models are based on the optimization of a \( F_\beta \) score for a classification of fingerprints distances closer than 4 m. However, most of the datasets have uniformly distributed fingerprints, where the pairwise distances are usually higher than that value. It has been observed that if all the distance pairs are allowed in the training, the model will focus more in the far region, where there is a smaller WiFi overlap and less correlation with the spatial distance. To minimize this effect, the training datasets only have fingerprint pairs where the pairwise distances are usually lower than that value. It has been observed that if all the distance pairs are allowed in the training, the model will focus more in the far region, where there is a smaller WiFi overlap and less correlation with the spatial distance. To minimize this effect, the training datasets only have fingerprint pairs where the distance is bellow 25 m, as we have found that increasing that threshold above that value affects the training, reducing the \( F_\beta \) score and increasing the error in the close region.

In addition to the numerical results, the comparison of the estimated and actual values for the three selected models is given in Fig. 9. By examining these figures, a deeper understanding of both the data structure and the distribution of estimated distances can be gained. As can be seen from the figure, especially for distances below 25 meters, the estimates obtained are very compatible with the actual distance. However, there is no estimation of models above this limit.
While we reported results with $F_{\beta}$ etc. We should note that our study mainly focused on metrics such as; Accuracy, Area under the ROC Curve (AUC), RMSE etc. We also calculated different $\beta$ value so this may cause lower achievement in other metrics. When comparing our results with other studies in the field, it is difficult to make an exact comparison. There are 2 main reasons for this; (1) Most of the studies use the data set obtained from the same building for testing and training purposes. (2) The studies give their results using different metrics such as; Accuracy, Area under the ROC Curve (AUC), RMSE etc. We should note that our study mainly focused on $F_{\beta}$ value so this may cause lower achievement in other metrics. While we reported results with $F_{\beta}$ we also calculated different metrics for comparison. The NearMe [42] study, one of the pioneering research in the field, carried out the training steps on the office building dataset and achieved the RMSE of 13.97 meters on the cafeteria building dataset as an unseen test-set. We achieved RMSE of 8.97 meters with XGBR algorithm on Huawei dataset, which is a reliable dataset in our experiment collected from shopping center and reflects the daily life which was used isolated datasets. Our models gives better results below 25 metres proximity but our test data demonstrates skewed distributions (more data above 25 metres), this causes an increase in the error rate. When we look at the results only focusing on 25 metres and below, we achieved quite improved results (see Table V). In the study named Wide [20], train and test sets were splitted from same building data and obtained 3.0 meters in terms of Mean absolute error (MAE) using deep neural network. Our study achieved MAE of 4.8 metres when evaluation was made of 25 meters or less between fingerprints with test data which has similar nature with train data. Also we got 5.4 metres in Huawei dataset. At first glance, the success rate seems low compared to the WIDE study, it is a good result in terms of being a generalizable score when we consider that our dataset is a composition of 11 different datasets from various environments. Another study [19], focuses only proximity inference task between two person and they reported 0.89 in AUC using 10 metres threshold. We achieved 0.907 in terms of $F_{\beta}$ score on Huawei dataset using XGBR with 4 metres distance.

TABLE V: Comparison of ML algorithms results for four datasets. Only fingerprint pairs with an actual distance of 25 meters or less between them were evaluated.

| ML   | MAE | RMSE | MSE  | Prec | Recall | $F_{\beta}$ |
|------|-----|------|------|------|--------|-------------|
| DTR  | 5.34 | 6.37 | 40.68 | 0.82 | 0.010 | 0.684 |
| LR   | 5.48 | 6.52 | 42.57 | 1.00 | 0.000 | 0.062 |
| KNNR | 6.20 | 7.21 | 53.01 | 0.76 | 0.004 | 0.446 |
| XGBR | 6.42 | 7.34 | 54.23 | 0.71 | 0.006 | 0.475 |
| LSVR | 6.42 | 7.27 | 53.15 | 0.76 | 0.005 | 0.483 |
| ANN  | 6.42 | 7.27 | 53.15 | 0.76 | 0.005 | 0.483 |

VIII. CONCLUSION

In our study, we developed a machine learning-based distance estimation model that does not depend on specific MAC vectors and does not require per venue training, which detects whether two fingerprints are close to each other and estimates the approximate distance between fingerprints. For this process, from the RSSI values we extracted 14 different features that represent relationships between fingerprint pairs using various difference metrics. After analyzing the importance score of these features, we found the best performing feature set using genetic algorithm. During these processes, we isolated the training and test data from each other to increase the robustness of the inference and to measure its extrapolation...
