3D Location Estimation using RSSI of Wireless LAN. A multi-technique approach with Machine Learning

Ndabuye S. Gideon
BSc. Software Engineering, Department of Computer Science and Engineering, The University of Dodoma.

ABSTRACT

In the dynamic landscape of wireless communication technologies, the accurate three-dimensional localization of receivers presents a persisting challenge. This work leverages the Received Signal Strength Indicator (RSSI) data obtained from a wireless LAN and uses advanced data processing and machine learning techniques to estimate the precise positions of receivers in three-dimensional space. The data preprocessing phase encompasses various critical steps, including RSSI value transformation, the integration of access point (AP) distances, time series data utilization, and robust outlier handling.

For modeling, the HistGradientBoosting algorithm is employed as the base estimator, to simultaneously predict receiver latitude, longitude, and height. The evaluation process assesses the localization error in 3D space.

In this work, two distinct training modes and dataset splits are used to gauge the impact of training data volume on model performance.

The results demonstrate noteworthy differences between the various training modes and dataset splits. Specifically, the "SET B" dataset split, utilizing a 75% training and 25% test split, yields the best average error of 0.45257 meters and a maximum error of 16.3630 meters. These outcomes reveal that data driven 3D location estimation is possible and its precision is proportional to the data available.

I. INTRODUCTION

In an era defined by the continuous evolution of wireless communication technologies, the need to precisely determine the location of a receiver in a three-dimensional space has become an increasingly pressing challenge. Various methods have been proposed, with the main aim of reducing the overall cost and resources in estimating the locations.

Various methods have been proposed to perform this task, some of these include distance-based, signal-based, direction-based and many others. Each having its own merits and demerits.

Note that, we could easily do this with GPS, but there two demerits that we are actually trying to avoid in that case. One being the issue of cost and resources in using GPS and the fact that GPS can be very inaccurate indoors, hence the exploration of other techniques. The WLAN positioning technique does not require additional hardware and can provide indoor positioning solely using an evenly distributed wireless network set up in the building and the smart device at the user's end [1].

In this work, we are going to utilize the Received Signal Strength Indicator (RSSI) to estimate the position of a receiver in a wireless LAN.

Received signal strength indicator (RSSI) is a measurement of the power present in a received radio signal. RSSI is usually invisible to a user of a receiving device. [2] Obviously, Wi-Fi is not designed or deployed for the purpose of positioning. However, the measurements of signal strength (SS) of the signal transmitted by either AP or station imply the possibility of finding the location of the mobile user (MU)

Knowing that there is no direct mathematical formula to get a location from only this information is why we employ data-driven techniques and machine learning to try and estimate the location in 3D.

We are going to employ various machine learning algorithms and evaluate how accurate they can be in estimating the position of a receiver in 3D.

II. METHOD

Data Description

As stated, in this work, we are going to use RSSI measured in dBm and other auxiliary information to estimate the 3D location of a receiver.

The dataset contains the following information.

- RSS information from transmitters whose positions are known. (These are the access points, APs)
- Information about the surrounding buildings such as RSSI measured with time stamp and SSID.
- Time-series GPS data (Latitude and Longitude) of the transmitter
- Channel setting, and map information of the measurement area to analyze LoS (Line of Sight)

The dataset includes only samples where there was a line of sight (LoS) between the access point (AP) and the receiver. Therefore, there are no physical obstacles for the signals.

The figure below shows the setup of the access points (APs) while recording the measurements.

Receivers are scattered around the area, both within the enclosed area and outside.

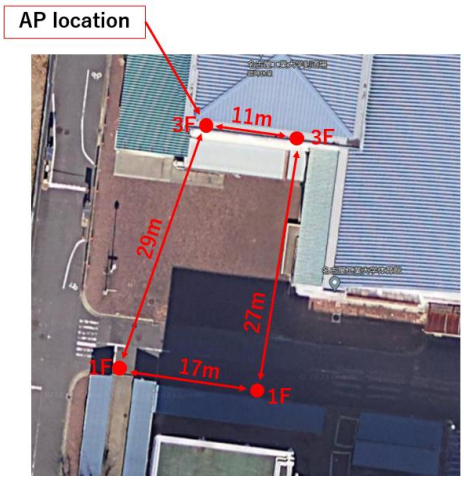


Figure 1: Location of APs

As it is a norm for machine learning, to make come out with a desirable solution, we need to have training data and evaluation data (test) where we'd see how well our developed models estimate the targets, in this case the 3D location of a receiver.

So, the data used was into train and evaluation sets. Let's take a look at the table below.

SET	**SIZE (rows)**
TRAIN	7,100
TEST	7,200
AP data	4

Each of the train and test sets had 8 columns, UnixTime, SSID, Frequency, Channel, RSSI, **Latitude, Longitude, Receiver_Height.** The ones in bold are our target, meaning we will build a machine learning model that will give the three given the other features.

AP Dataset contained the latitudes and longitudes of the 4 APs.

Let's first visualize the location distribution of the APs along with the receivers in a scatterplot.

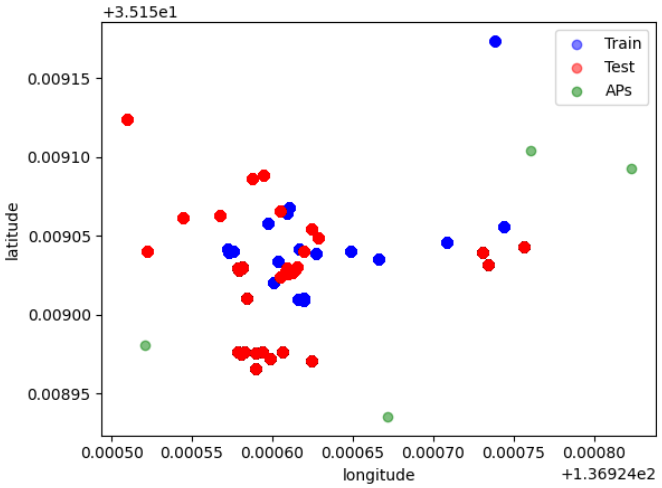


Figure 2: Scatterplot showing the location distribution of Train and Test samples along with the APs

Data Preprocessing

Also, strongly, as a norm, we need to process and features engineer the data to a desired form such that is provides mostly useful information in estimation of our target.

This was done in the four below categories.

1. RSSI

One of the main features that has a strong correlation with the 3D position of the receiver is the RSSI. Strong attention was put into this feature.

Values of RSSI are mostly from -100dBm to 0 or 100dBm in some devices. From the dataset, values ranged from -80 to -44. Now, for machine learning, this kind of range of values don't play well,

We need to transform the RSSI values to a more standard and well distributed way.[3] In their work, showed that three RSSI transformations, ZeroToOneNormalized (RSSI), Exponential (RSSI) and Power (RSSI) gave a significant boost in performance of their work.

Our experimental observations show that *ZeroToOneNormalized (RSSI)* and *Power (RSSI)* are more significant based on the data we have. These are given as shown below.

$$ZeroToOneNormalized(RSSI) = \frac{RSSI - c}{-c}$$

$$Powed(RSSI) = \left(\frac{RSSI - c}{-c}\right)^{\beta}$$

In this case c is the value equal to the minimum RSSI value and β is a constant, is this case, the mathematical constant e.

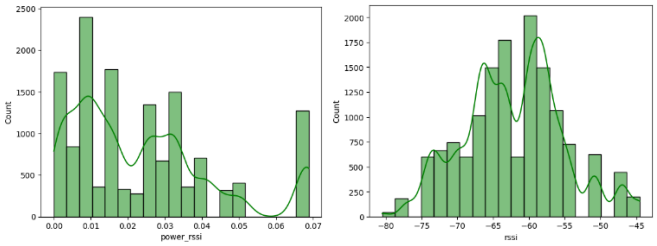


Figure 3: Comparing RSSI and the Power transformation

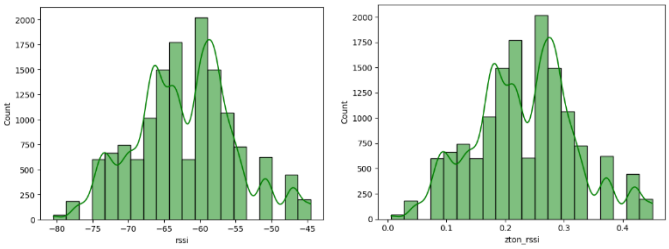


Figure 4: Comparing RSSI and the Zero-to-One Normalized RSSI

Since the RSSI is affected by many factors, like distance, channel settings and many more, the *Channel* and *APs* were pivoted so that each could represent its own value of the two generated transformations of RSSI.

2. Access Points (APs)

To ensure that we also give the model as much information as possible, another vital data was the distances between all the available APs in the area. This should give the model relative information that correlates with the RSSI and hence have information about the target location.

3. Time Series info

Given the time series information that show a range 3 hours. It was not useful to use hours or days as they wouldn't give us much information. Therefore, since we are talking about radio waves, the difference in seconds or less would be very useful.

Thus, a lag of 1 step was used in seconds to match the records step. This should give us some relationship with the Time of Arrival of the signal.

4. Miscellaneous

To ensure that we don't have data points beyond normal expected measures, outliers were replaced with the upper and lower boundaries.

This ensures that the range of values we go and pass to the machine learning algorithm are well within expected range.

Modelling

1. Algorithm

Since we are estimating three values of very different range, a multi-model regression algorithm is the best option. Latitudes have their own range, so do longitudes and likewise with the heights.

In this case, proposed a gradient boosting algorithm known as HistGradientBoosting algorithm as the base estimator.

This is because it runs by mostly **binning** features which is the default nature of our data (looking continuous but in some way categorical), it also handles missing values. So, this algorithm fits this particular purpose.

As a base estimator, we put this into a multi-output wrapper from sklearn that does the job.

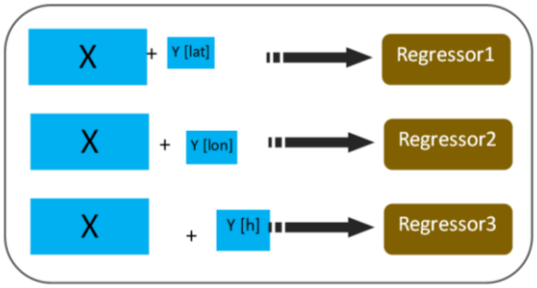


Figure 5: Model training mechanism

2. Training

The base estimator, HistGradientBoostingRegressor was first wrapped on a grid search for perfect hyperparameters for each of the target values.

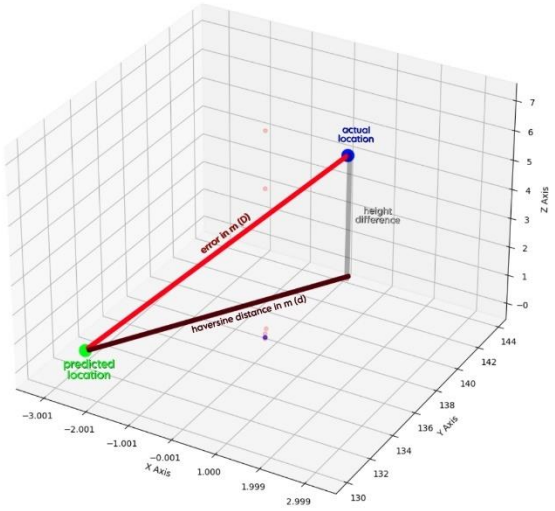
After hyperparameter tuning, two training methods were employed and both reported for deeper analysis.

1. Trained on a single shot of train set
2. Train using Stratified Kfold, while stratifying on the target '***receiver_height***'.

On each of the two methods, there were two different sizes of train and evaluation sets as we will see in the results section

3. Evaluation Method

Evaluation of the model's predictions were done in the localization error in meters, in 3D space. It is done in two steps; we first calculate the actual distance in meters using the Haversine formula and then employ the Pythagoras theorem to get the actual distance in 3D from the heights.



To get the error, D, we first calculate the haversine distance (d) using the latitude and longitudes using the haversine formula. Here R=6378137, the radius of earth in meters.

$$a = \sin^2\left(\frac{\Delta \text{lat}}{2}\right) + \cos(\text{lat}_1) \cdot \cos(\text{lat}_2) \cdot \sin^2\left(\frac{\Delta \text{lon}}{2}\right)$$

$$c = 2 \cdot \text{atan2}\left(\sqrt{a}, \sqrt{1-a}\right)$$

$$d = R \cdot c$$

$$b = \text{height}_2 - \text{height}_1$$

$$D_{3D} = \sqrt{D_{2D}^2 + b^2}$$

In this case, the error in distance (D) acts as a Hypotenuse in a right-angled triangle. This is our distance in 3D space. These are shorter distances, so the curvature of the earth can be neglected.

III. Results and Discussion

Overall Performance and Complexity

To get a good ground for evaluation, two different training modes and sets were used, thus giving out four different results and models.

The table below shows the general performance of the models as measured in average error and maximum error in meters (m).

1. SET A: Designated Train and Evaluation Sets

Train Mode	Average Error (m)	Max Error (m)
Single Batch	**2.7244**	**16.3086**
6 Stratified Fold	**2.8285**	**16.3086**

2. SET B: 75% by 25% Train and Test Split

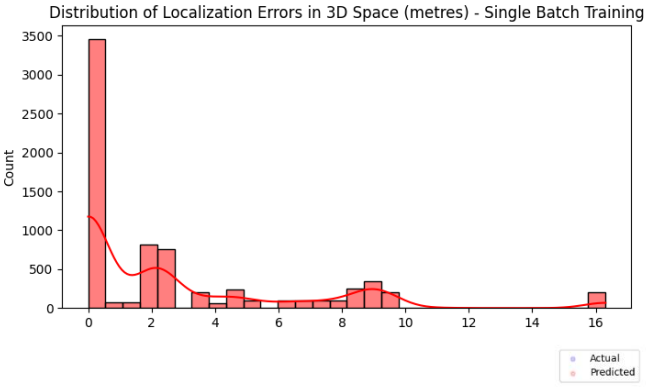
Train Mode	Average Error (m)	Max Error (m)
Single Batch	**0.53643**	**16.06050**
6 Stratified Fold	**0.47332**	**16.05676**

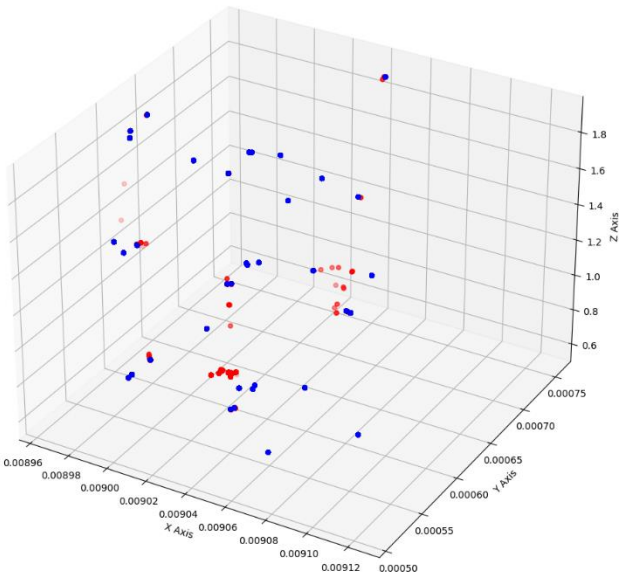
We can see there is an interesting difference, it's obvious that with more training data, the model should have a better performance on unseen data compared to when trained on less. But whats eye-catching here is that, for set A the single batch-trained model does better slightly better in average error but they appear the same in the max error. For set B, the single-batch trained model does well in max error but less in average error. A final fact we observe is that the maximum error has remained almost the same in all four cases. Let's see how this looks visually.

Visualizing the predictions

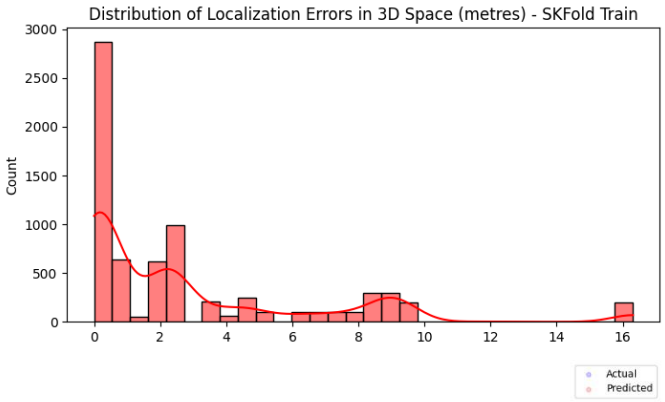
At this point, it's quite necessary to visualize and see the distribution of the errors the model makes in estimating the 3D location. This should tell us about the 16m max error, as in whether it's the only values of the values remain the same.

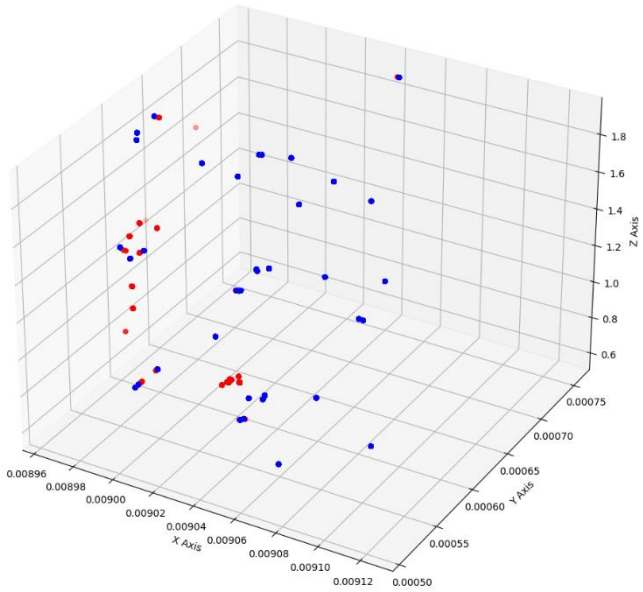
1. SET A: Designated Train and Evaluation Sets





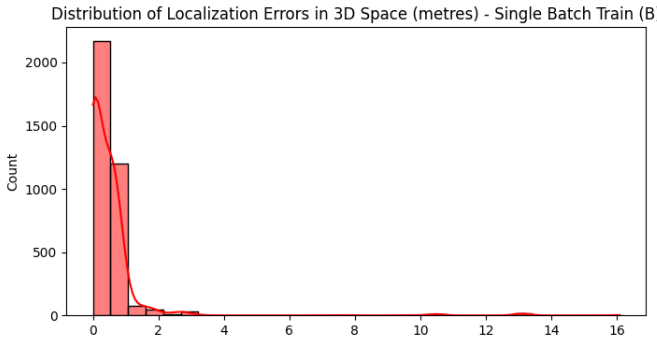
From this single batch training, we can see many samples with an error of around 16 meters also between 8 and 10. But most of them lie around zero, which is a good thing.

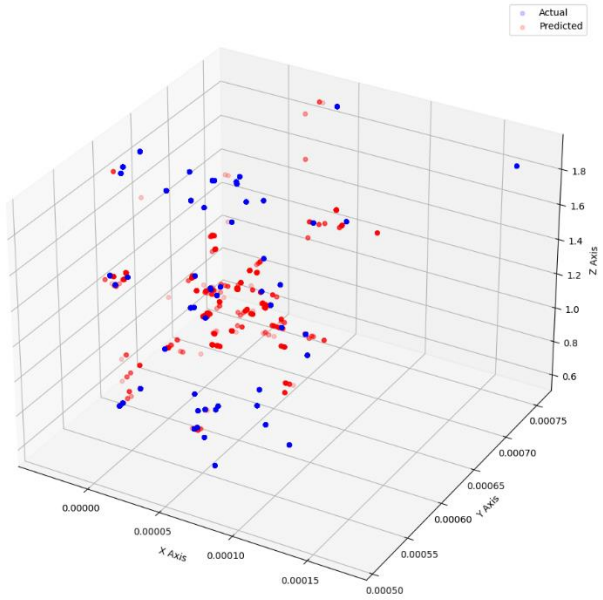




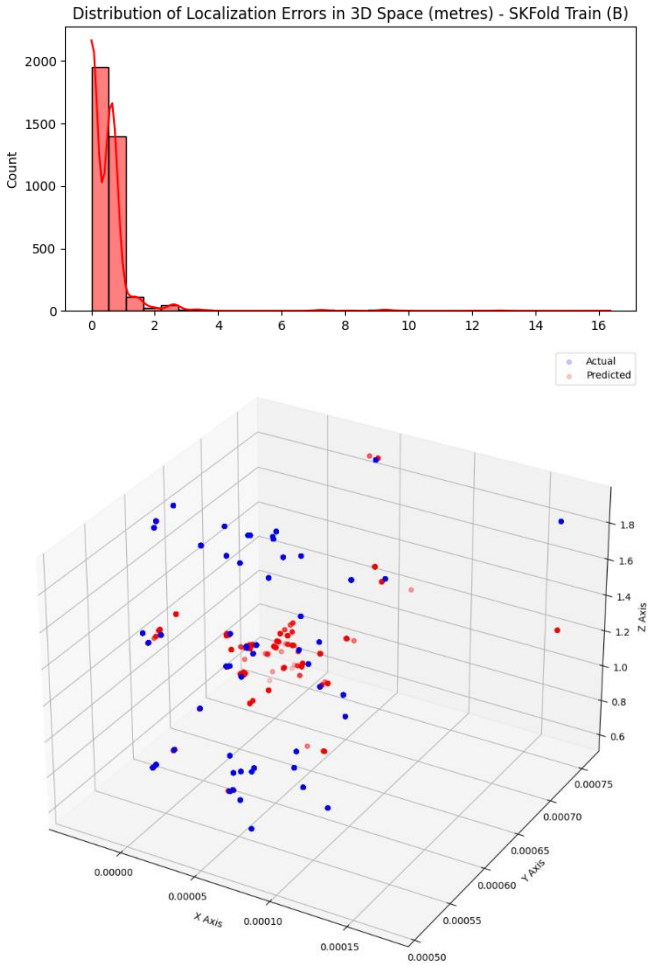
The overall performance here is lower compared when trained on a single batch above. Here, we see the first bar struggling to reach 3000 while the previous was almost at 3500, meaning that with a single batch training, the model has more predictions whose errors approach zero.

2. SET B: 75% by 25% Train and Test Split





Here, we see a very small number of samples with errors greater than 4, the 3D plot shows less overlap but the points are closer than upper plots.



The model when trained on Stratified Kfold in this set, tends to have a better average score than the one with a single shot train.

IV. Conclusion

In conclusion, this work addresses the challenge of precisely determining the three-dimensional location of a receiver in the context of evolving wireless communication technologies. The study employs the Received Signal Strength Indicator (RSSI) in a wireless LAN to estimate receiver positions and utilizes machine learning techniques for this purpose.

The key contributions of this work include data preprocessing, where RSSI values are transformed for better model performance, and the incorporation of distances between access points (APs) to provide additional context for the model. Time series information and outlier handling were also crucial steps in preparing the data for modeling.

The choice of the HistGradientBoosting algorithm as the base estimator in a multi-model regression framework proved effective in estimating latitude, longitude, and receiver height simultaneously. The model's performance was evaluated in terms of localization error in 3D space, with results showing differences between different training modes and dataset splits.

Enhancements in data collection and feature engineering methods could also contribute to better results if obtain more data. Finally, this work could be extended to address the challenges of indoor localization, which remains a significant issue in wireless communication technologies.

V. References

[1] L. Wang, S. Shang, and Z. Wu, “Research on Indoor 3D Positioning Algorithm Based on WiFi Fingerprint,” *Sensors*, vol. 23, no. 1, Jan. 2023, doi: 10.3390/s23010153.

[2] B. Li, I. J. Quader, and A. G. Dempster, “On outdoor positioning with Wi-Fi,” *Journal of Global Positioning Systems*, vol. 7, no. 1, pp. 18–26, Jun. 2008, doi: 10.5081/JGPS.7.1.18.

[3] M. Nurpeiissov, A. Kuzdeuov, A. Assylkhanov, Y. Khassanov, and H. A. Varol, “End-to-End Sequential Indoor Localization Using Smartphone Inertial Sensors and WiFi,” *2022 IEEE/SICE International Symposium on System Integration, SII 2022*, pp. 566–571, 2022, doi: 10.1109/SII52469.2022.9708854.