A Comparison of RNN, LSTM and GRU on Indoor Localization using BLE RSSI Measurements

Gokce Gok

Graduate School of Natural And Applied Sciences
Dokuz Eylul University
gok.gokce@ogr.deu.edu.tr

In this study, I compare RNN, LSTM and GRU methods for indoor localization/navigation with BLE RSSI readings.

DATASET

The dataset was created using the RSSI readings of an array of 13 iBeacons in the first floor of Waldo Library, Western Michigan University. Data was collected using iPhone 6S. The dataset contains two sub-datasets: a labeled dataset (1420 instances) and an unlabeled dataset (5191 instances). The recording was performed during the operational hours of the library. For the labeled dataset, the input data contains the location (label column), a timestamp, followed by RSSI readings of 13 iBeacons. RSSI measurements are negative values. Bigger RSSI values indicate closer proximity to a given iBeacon (e.g., RSSI of -65 represents a closer distance to a given iBeacon compared to RSSI of -85). For out-of-range iBeacons, the RSSI is indicated by -200. The locations related to RSSI readings are combined in one column consisting of a letter for the column and a number for the row of the position[1].

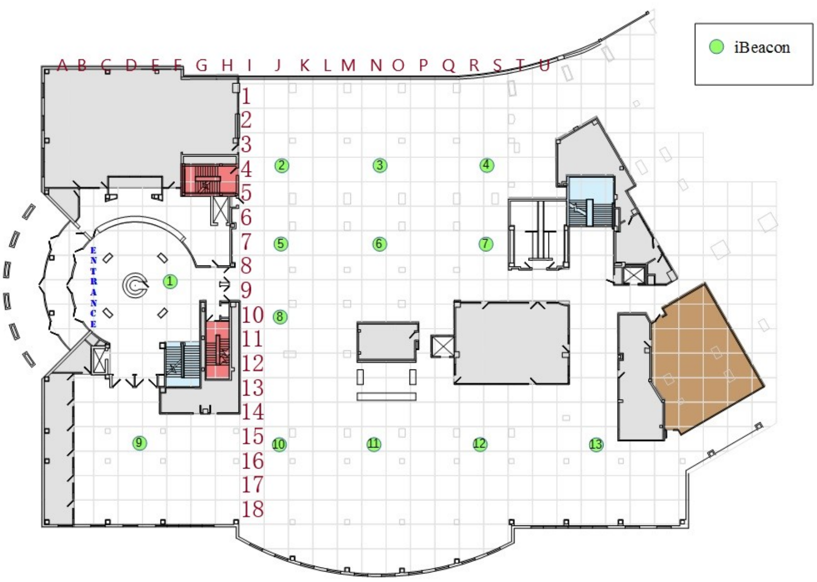

Figure 1. The Layout of the iBeacons.

In the original dataset, locations are combinations of letters and numbers. Letters symbolize the x-axis and the numbers the y-axis. I translated the letters to numbers for using them like cartesian coordinates.

Before, the locations looked like A01, B05, I02 etc. I transformed this symbolization to like (0, 1), (1, 5), (8, 2) etc. So, I treated the problem as a regression problem and tried to solve it from that perspective.

location	x_coor	y_coor
I01	8	1
I01	8	1
I01	8	1
I01	8	1
I01	8	1

Figure 2. Head of location part of dataset

Before the model building part, I scaled data with Min-Max Scaler. And, split scaled data to training, test and validation sets. Firstly, I split test data as 15% of all data. After that, split validation data as %10 of training data.

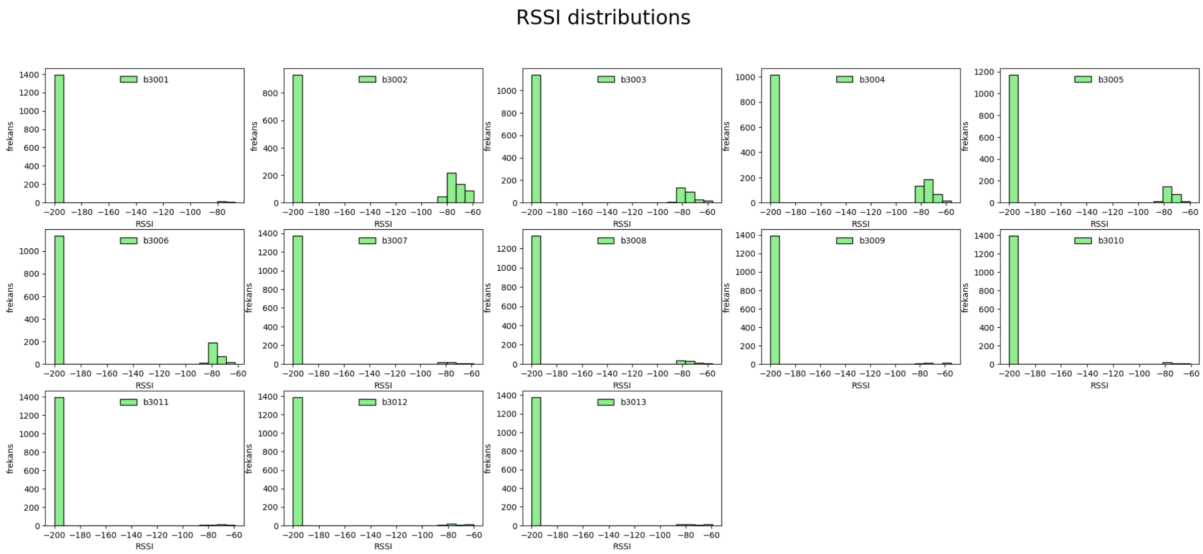

Figure 3. Distribution of RSSI measurements in iBeacons

PERFORMANCE METRICS

As I mentioned before, I generate x and y cartesian coordinates from the “location” feature. Therefore, I adapted the performance metrics to include a geometric perspective. I used Euclidean Distance for RMSE calculation and Manhattan Distance for MAE calculation. I calculated the MAPE separately in the x and y directions.

NUMBER OF PARAMETERS & RUNNING TIME

I built the models with the same number of layers and units for a healthy comparison.

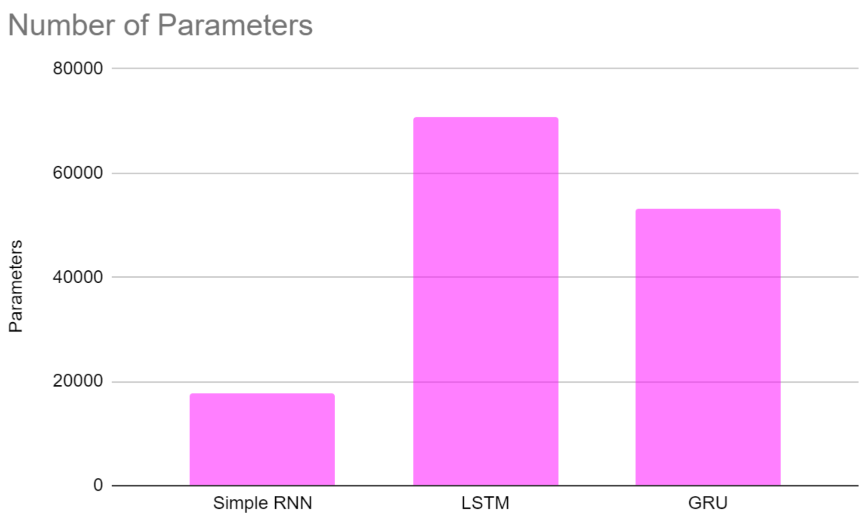

Figure 4. Comparison of model parameters

I write my codes in Google Colab with Python. I performed the training using both CPU and TPU. The chart below shows a comparison of the models in runtime. Actually, it is not correct to make this comparison because the number of parameters changes.

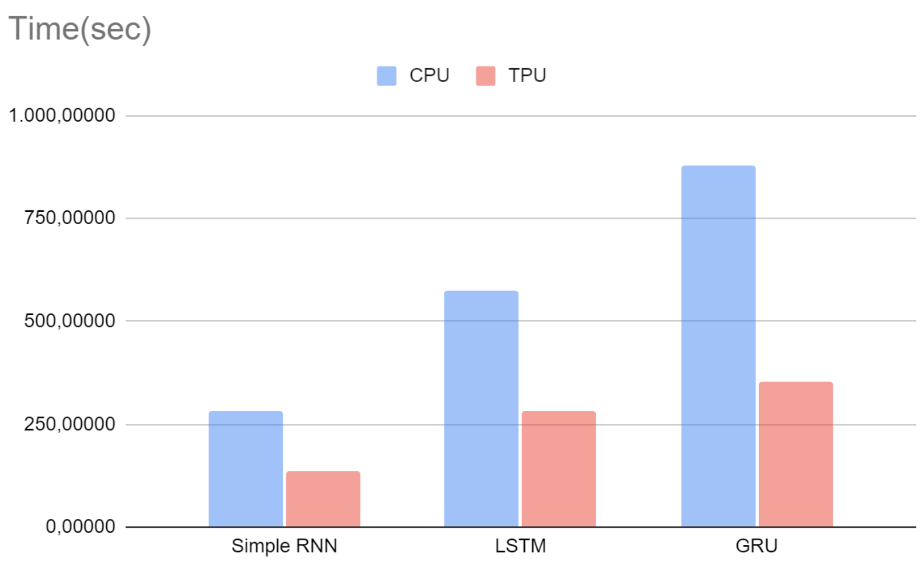

Figure 5. Comparison of runtime

I complete the training and validation part after 300 epochs in all models.

RNN MODEL

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 simple_rnn_16 (SimpleRNN)   (None, 13, 64)            4224

 dropout_22 (Dropout)        (None, 13, 64)            0

 simple_rnn_17 (SimpleRNN)   (None, 13, 64)            8256

 dropout_23 (Dropout)        (None, 13, 64)            0

 simple_rnn_18 (SimpleRNN)   (None, 13, 32)            3104

 dropout_24 (Dropout)        (None, 13, 32)            0

 simple_rnn_19 (SimpleRNN)   (None, 32)                2080

 dropout_25 (Dropout)        (None, 32)                0

 dense_6 (Dense)             (None, 2)                 66

=================================================================
Total params: 17,730
Trainable params: 17,730
Non-trainable params: 0
```

Figure 6. Architecture of RNN

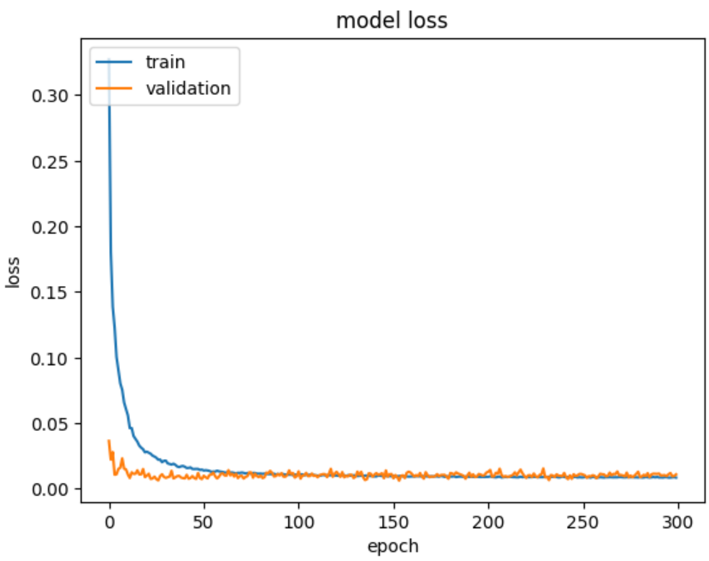

Figure 7. Model Loss at RNN

	Size	R-squared	Pearson Coef	RMSE	MAE	MAPE (x)	MAPE(y)
Train	1086	0,89620	0,94670	0,10401	0,13192	2,6645	1534,50838
Validation	213	0,65789	0,91488	0,13712	0,17977	0,1363	2,66451
Test	121	0,89050	0,94485	0,10693	0,13617	0,1309	1913,34346

Table 1. Performance Metrics of RNN

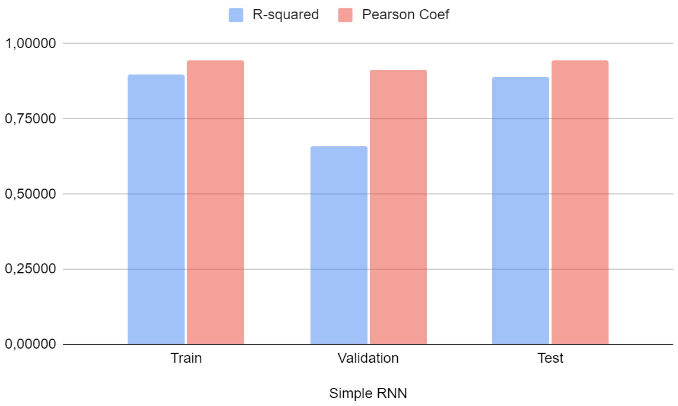

Figure 8. R-squared and Pearson Coef metrics of RNN

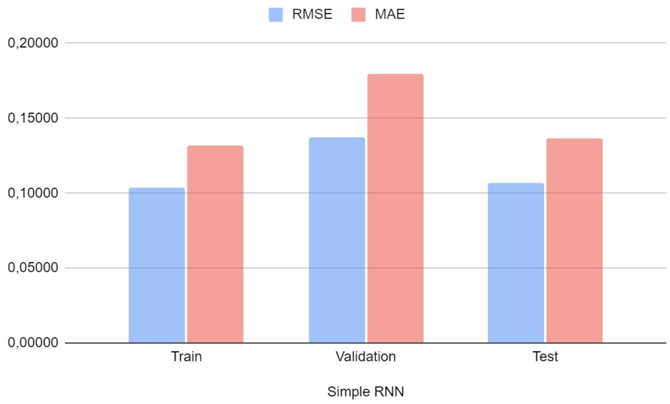

Figure 9. RMSE and MAE metrics of RNN

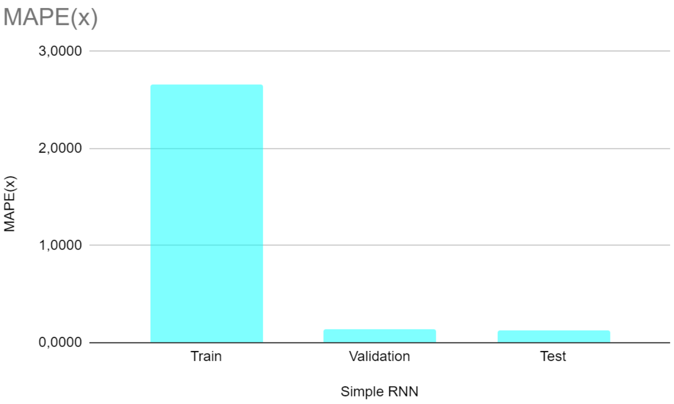

Figure 10. MAPE(Horizontal) of RNN

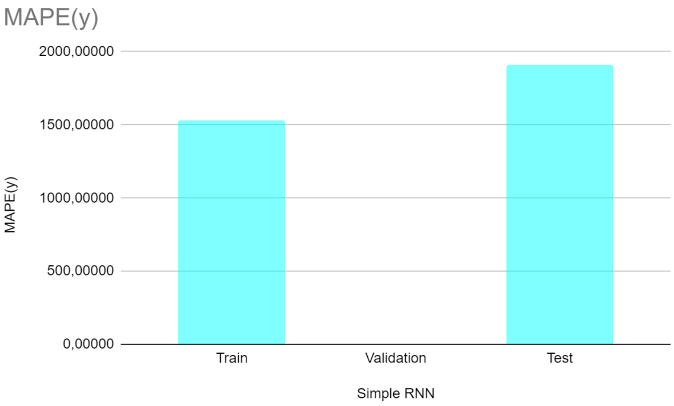

Figure 11. MAPE(Horizontal) of RNN

LSTM MODEL

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_8 (LSTM)               (None, 13, 64)            16896

 dropout_26 (Dropout)        (None, 13, 64)            0

 lstm_9 (LSTM)               (None, 13, 64)            33024

 dropout_27 (Dropout)        (None, 13, 64)            0

 lstm_10 (LSTM)              (None, 13, 32)            12416

 dropout_28 (Dropout)        (None, 13, 32)            0

 lstm_11 (LSTM)              (None, 32)                8320

 dropout_29 (Dropout)        (None, 32)                0

 dense_7 (Dense)             (None, 2)                 66

=================================================================
Total params: 70,722
Trainable params: 70,722
Non-trainable params: 0
```

Figure 12. Architecture of LSTM

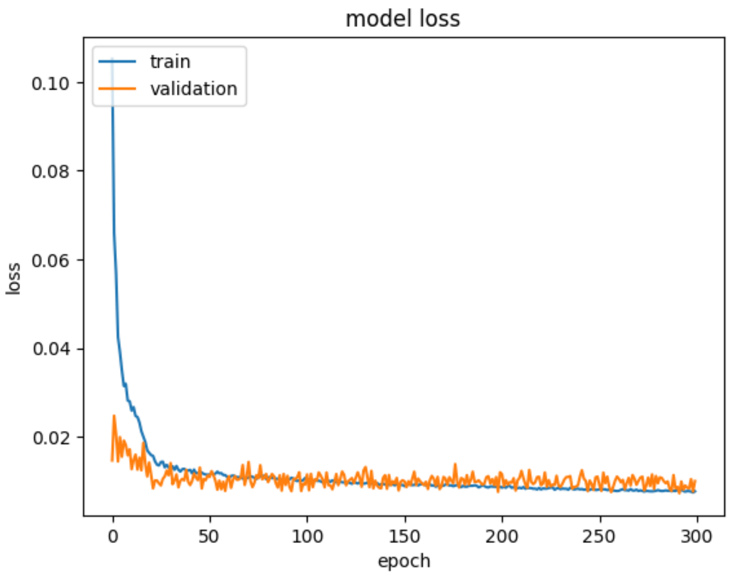

Figure 13. Model Loss at RNN

	Size	R-squared	Pearson Coef	RMSE	MAE	MAPE (x)	MAPE(y)
Train	1086	0,90461	0,95148	0,09724	0,12205	17,0129	1409,87877
Validation	213	0,70549	0,92721	0,12235	0,16069	0,1171	0,30229
Test	121	0,88913	0,94827	0,10781	0,13507	0,1320	1755,10622

Table 2. Performance Metrics of LSTM

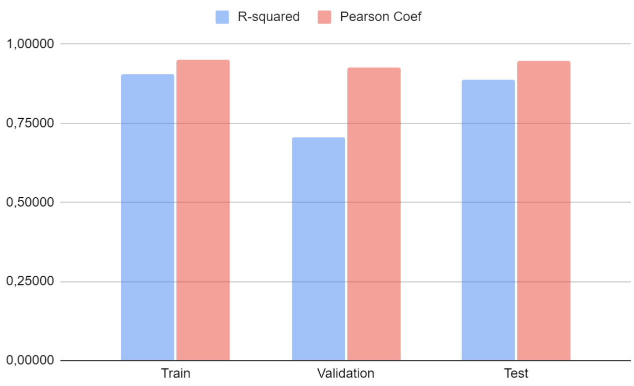

Figure 14. R-squared and Pearson Coef metrics of LSTM

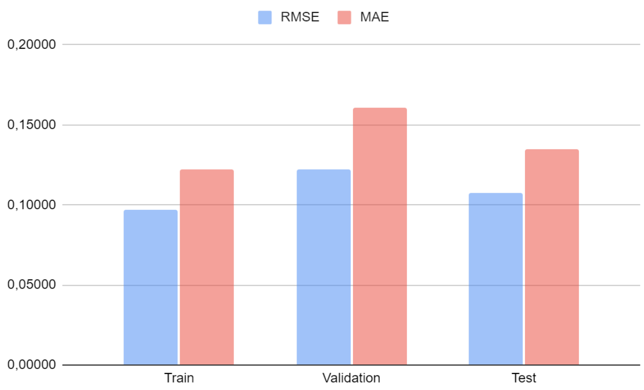

Figure 15. RMSE and MAE metrics of LSTM

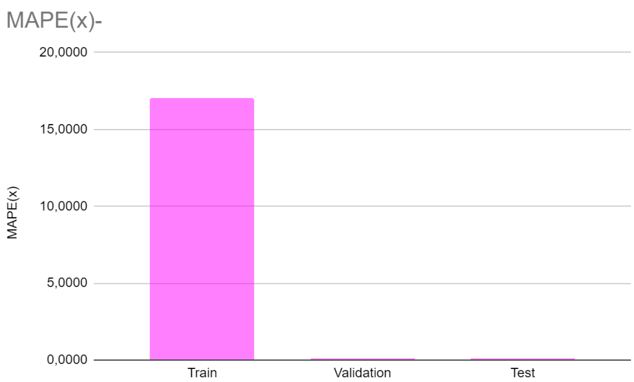

Figure 16. MAPE(Horizontal) of LSTM

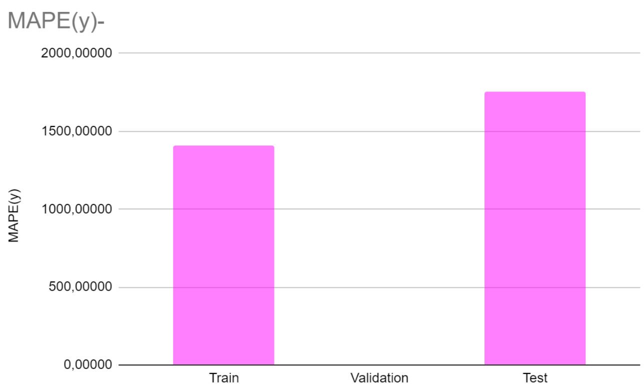

Figure 17. MAPE(Horizontal) of LSTM

GRU MODEL

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 gru (GRU)                   (None, 13, 64)            12672

 dropout_30 (Dropout)        (None, 13, 64)            0

 gru_1 (GRU)                 (None, 13, 64)            24768

 dropout_31 (Dropout)        (None, 13, 64)            0

 gru_2 (GRU)                 (None, 13, 32)            9312

 dropout_32 (Dropout)        (None, 13, 32)            0

 gru_3 (GRU)                 (None, 32)                6240

 dropout_33 (Dropout)        (None, 32)                0

 dense_8 (Dense)             (None, 2)                 66

=================================================================
Total params: 53,058
Trainable params: 53,058
Non-trainable params: 0
```

Figure 18. Architecture of GRU

Figure 19. Model Loss at GRU

	Size	R-squared	Pearson Coef	RMSE	MAE	MAPE (x)	MAPE(y)
Train	1086	0,90324	0,95053	0,09879	0,12491	74,5717	1577,82286
Validation	213	0,75862	0,92513	0,10936	0,14269	0,1164	0,24667
Test	121	0,87847	0,94212	0,11158	0,13937	0,1342	2004,85453

Table 3. Performance metrics of GRU

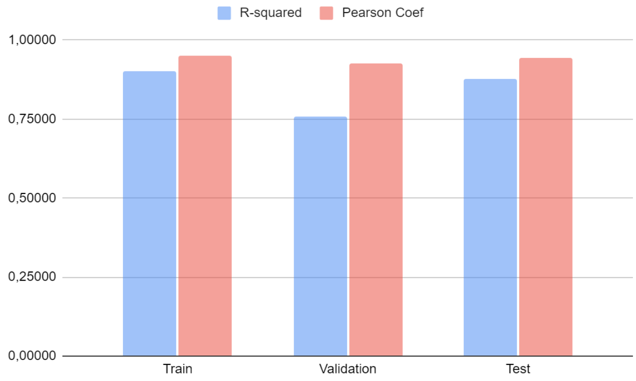

Figure 20. R-squared and Pearson Coef metrics of GRU

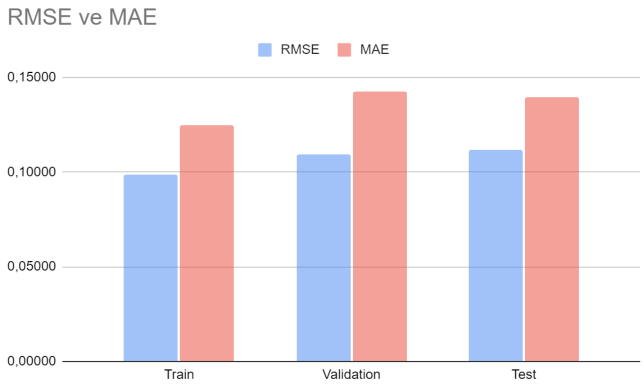

Figure 21. RMSE and MAE metrics of GRU

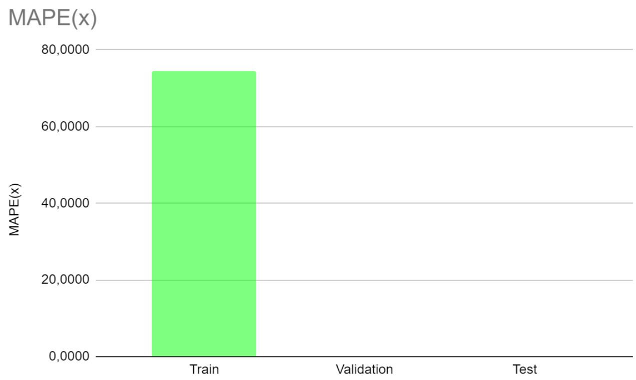

Figure 22. MAPE(Horizontal) of GRU

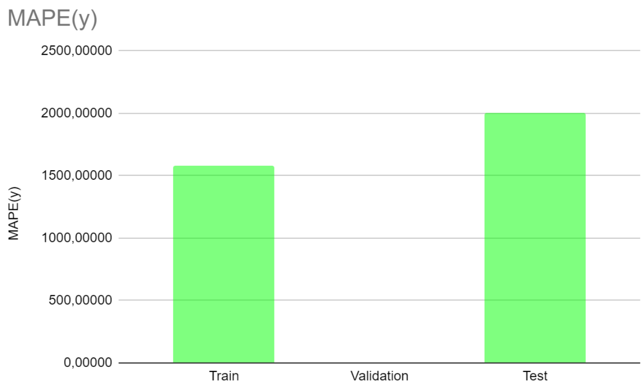

Figure 23. MAPE(Horizontal) of GRU

COMPARISON OF METRICS WITH TEST DATA

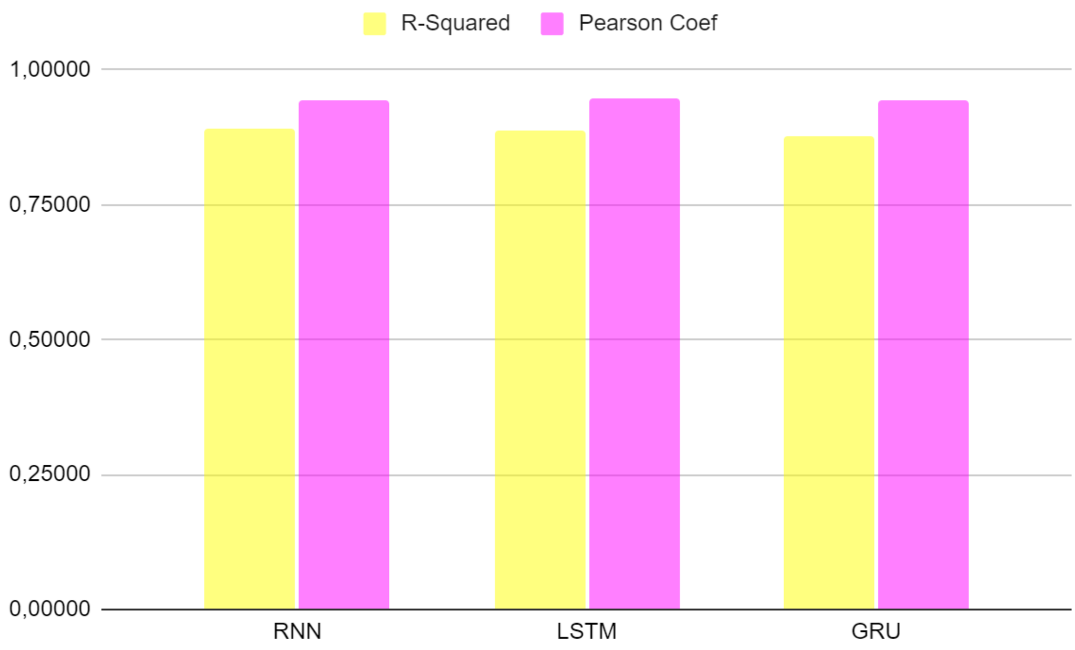

Figure 24. Comparison of R-squared and Pearson Coef Metrics of Models

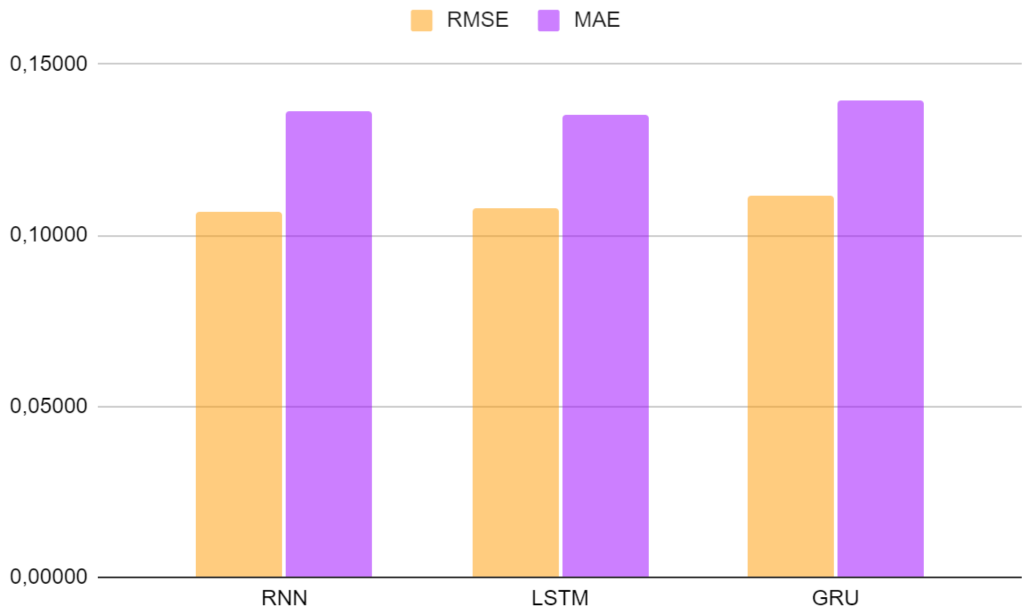

Figure 25. Comparison of RMSE and MAE Metrics of Models

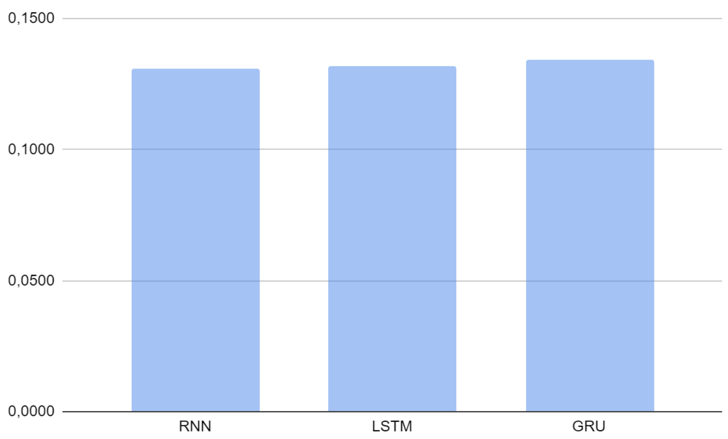

Figure 26. Comparison of MAPE Metrics in Horizontal Coordinates of Models

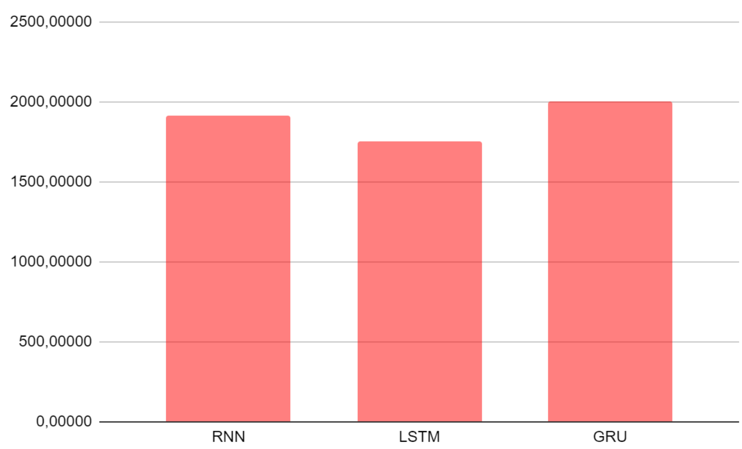

Figure 27. Comparison of MAPE Metrics in Vertical Coordinates of Models

In conclusion, when all metrics are evaluated separately for test, training and validation sets, the best results were obtained with LSTM.

VISUALIZATION OF ACTUAL LOCATIONS AND PREDICTIONS

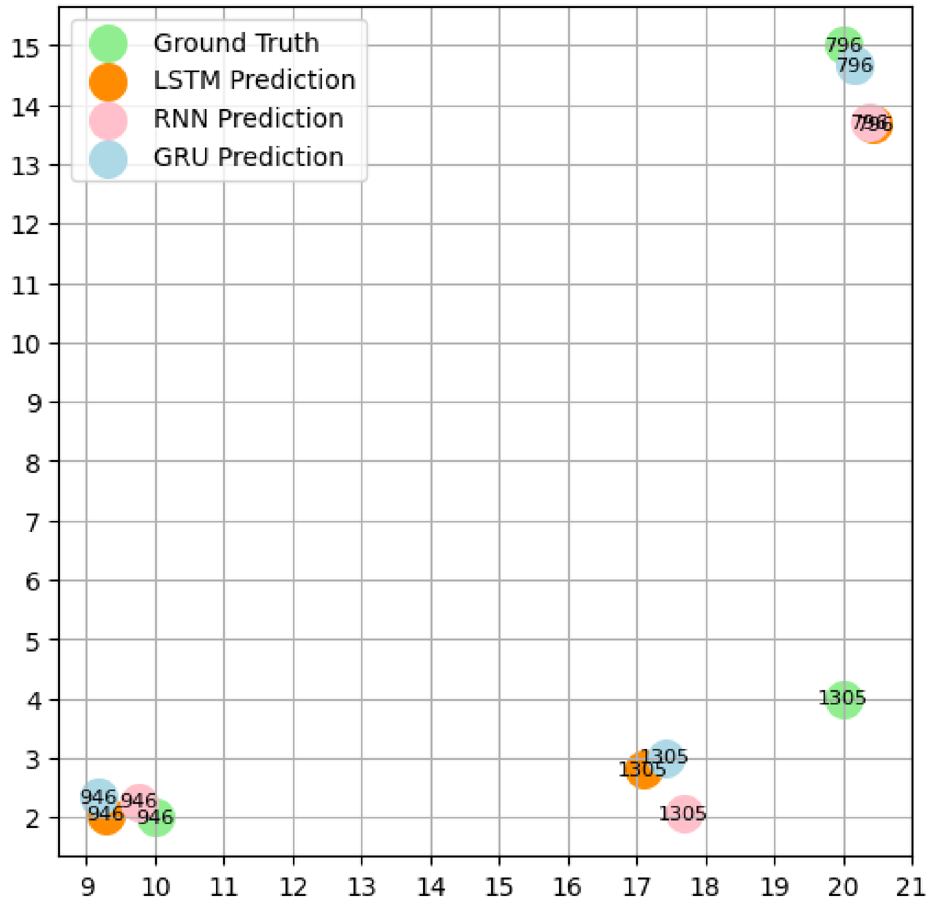

Figure 28. Visualization of Randomly Selected 3 Points

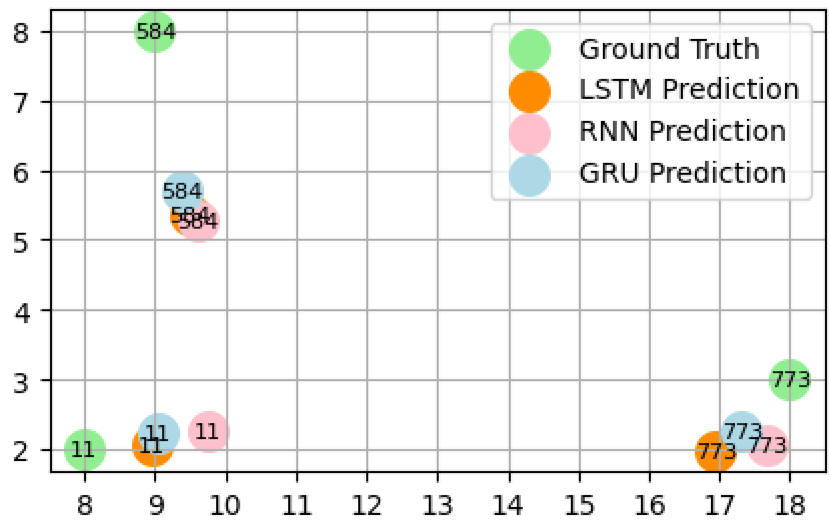

Figure 29. Visualization of Randomly Selected 3 Points

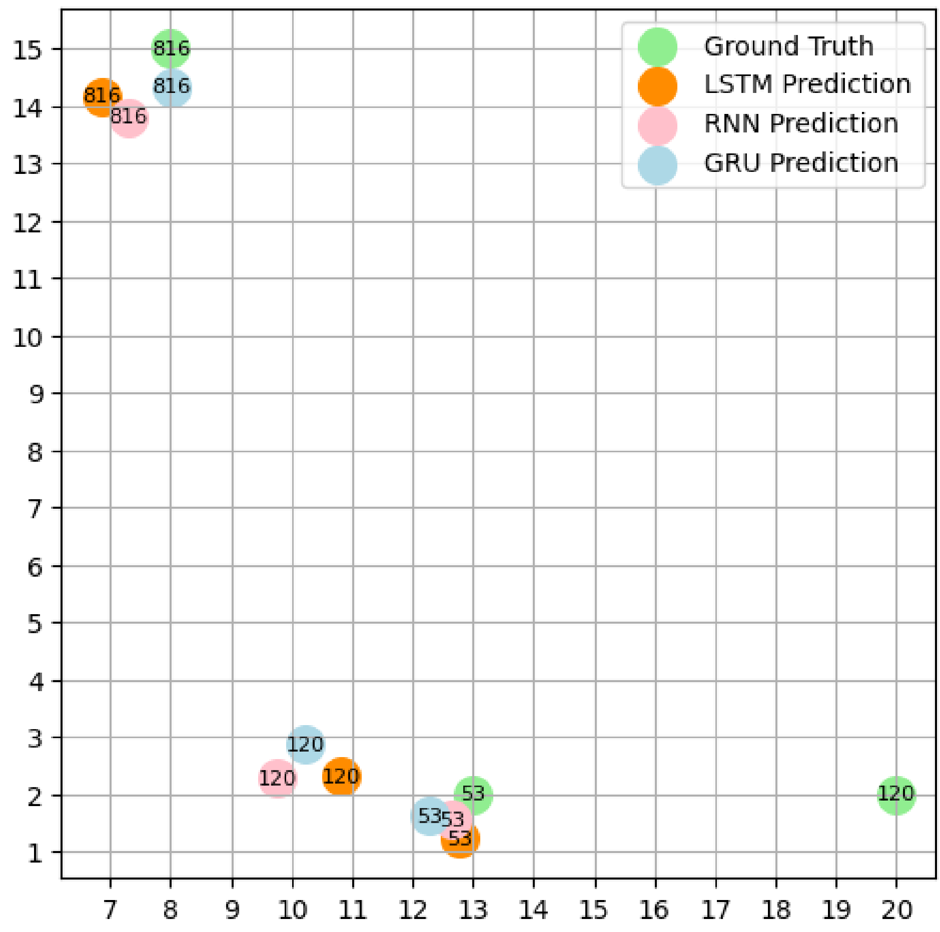

Figure 30. Visualization of Randomly Selected 3 Points

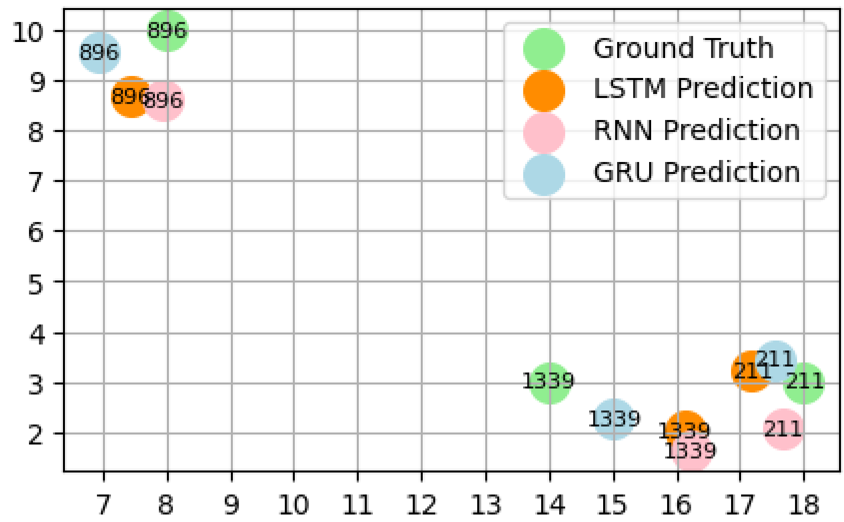

Figure 31. Visualization of Randomly Selected 3 Points

REFERENCES

[1] M. Mohammadi, BLE RSSI Dataset for Indoor Localization and Navigation

[2] I. Alexander, G. P. Kusuma, "Predicting Indoor Position Using Bluetooth Low Energy and Machine Learning", International Journal of Scientific \& Technology Research Vol 8, 2019.

[3] S. Akleylek,, E. Kılıç,, B. Söylemez,, T.E. Aruk,, A. Çavuş Yıldırım,, "A Research On Indoor Positioning Systems", Journal of Engineering Sciences and Design, 8(5), 90-105

[4] R. Ramirez,, C.-Y. Huang,, C.-A. Liao,, P.-T. Lin,, H.-W. Lin,, S.-H. Liang,, "A Practice of BLE RSSI Measurement for Indoor Positioning". Sensors 2021, 21, 5181. https://doi.org/10.3390/s21155181

[5] Z. Jianyong, L. Haiyong, C. Zili and L. Zhaohui, "RSSI based Bluetooth low energy indoor positioning," 2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Busan, 2014, pp. 526-533, doi: 10.1109/IPIN.2014.7275525

[6] C. Jain, G. V. S. Sashank, V. N and S. Markkandan, "Low-cost BLE based Indoor Localization using RSSI Fingerprinting and Machine Learning," 2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 2021, pp. 363-367, doi: 10.1109/WiSPNET51692.2021.9419388