An ML-Based Location Tracking System for LoRa Mesh Networks in GPS-Denied Environments

Sheikh Tareq Ahmed
Electrical & Computer Engineering
Prairie View A&M University
Prairie View, USA

Annamalai Annamalai
Electrical & Computer Engineering
Prairie View A&M University
Prairie View, USA

Ahmed Abdelmoamen Ahmed
Computer Science
Prairie View A&M University
Prairie View, USA

Mohamed Chouikha
Electrical & Computer Engineering
Prairie View A&M University
Prairie View, USA

Sanjaya Subedi
Computer Science & Engineering
Howard University
Washington, DC, USA

Miguel Polanco
Independent Researcher
Houston, Texas, USA

***Abstract*—LoRaWAN technology is a cornerstone in developing low-power wireless communication solutions for smart cities, agriculture, and remote environmental monitoring applications. LoRa enables long-range communication with minimal power usage, making it a suitable technology for battery-operated IoT devices. However, LoRa mesh networks rely on GPS to identify the location of its LoRa End Devices (EDs), which may be unreliable or unavailable in GPS-denied environments. This paper presents a location tracking and prediction system for LoRa mesh networks using Machine Learning (ML). In particular, we evaluate the performance of various ML models for predicting the geographic coordinates of LoRa EDs across the network. We trained four ML models, namely Linear Regression, Random Forest, KNeighbors Regressor, and Decision Tree Regressor, with a dataset collected by conducting real-world testbed and physical LoRaWAN hardware. Comprehensive hyperparameter tuning was conducted to enhance model performance. Our experimental results demonstrate that Random Forest, with its ability to capture complex and non-linear relationships, significantly outperforms other models in terms of accuracy and robustness. While lightweight ML models like Linear Regression offer competitive performance, they fall short in explanatory power.**

***Index Terms*—LoRa, LoRaWAN, GPS, Machine Learning (ML), Remote Monitoring, IoT.**

I. Introduction

The efficient geolocation of mobile objects and individuals plays a pivotal role in modern tracking systems in various fields, including agriculture, healthcare, military, logistics, wildlife, and public safety [1]–[4]. However, traditional GPS-based tracking systems often falter when GPS signals are obstructed or absent [5], such as indoor buildings, underground facilities, or densely constructed urban areas [6], [7]. This presents a critical need for alternative tracking methodologies that remain functional irrespective of satellite signal availability, thus ensuring continuous and reliable location tracking [8].

This paper proposes a location tracking system that utilizes the Long Range Wide Area Network (LoRa) mesh networks [2] in GPS-denied environments. With its low power consumption and ability to facilitate long-range communication, LoRaWAN stands out as an optimal solution for indoor and outdoor tracking systems [9]. The network's efficiency in transmitting low data rate signals over considerable distances makes it suitable for applications where energy efficiency and network coverage are of paramount importance [10].

With the advancements of Machine Learning (ML) [6], it is feasible to predict the geographic coordinates of LoRa End Devices (EDs) in a LoRa mesh network with high accuracy without using GPS modules. Four different ML models, namely Linear Regression, Random Forest, K-Neighbors Regressor, and Decision Tree Regressor, were trained with a dataset generated from a real-world testbed conducted using physical LoRaWAN hardware.

The testbed, which consisted of three static anchor LoRa nodes and one mobile node, addressed several challenges, such as ensuring consistent signal reception and mitigating environmental interference. The three anchor nodes were strategically placed approximately two miles apart, forming a unilateral triangle. This configuration is essential for triangulating the position of a mobile node that moves within the perimeter of these anchor nodes. By integrating LoRaWAN with a mesh topology, the experiment leverages the redundancy and connectivity benefits of mesh networking, enhancing the reliability and robustness of the data transmission.

Hyperparameter tuning was essential to optimize model performance, balancing complexity, avoiding overfitting, and achieving computational efficiency. The ML models are used to refine the accuracy and reliability of location estimations within the LoRaWAN network. The ML models meticulously analyze various data types collected from the network, including Received Signal Strength Indicators (RSSI), Signal-to-Noise Ratio (SNR), and Time Difference of Arrival (TDoA), along with the historic recorded latitude and longitude for initial calibration purposes. By training the ML models with a dataset from ground truth EDs' locations, the system can predict the position of the mobile EDs based on the unique patterns and strengths of the received signals.

II. Related Work

Location tracking in GPS-denied environments, such as indoor settings, urban canyons, and underground areas, has garnered significant research attention [5], [11], [12] due to the limitations of traditional GPS-based systems, which typically provide high accuracy in open environments but often falter when signals are obstructed or unavailable [6]. Alternative localization methods have been developed to address these challenges, with Wireless Sensor Networks (WSNs) emerging as a key technology in this domain [1].

The trilateration-based positioning model [5] tried to estimate the location of WSN nodes without GPS by leveraging the Received Signal Strength (RSS) from multiple anchor nodes. It demonstrated efficacy in indoor environments with obstacles and achieved low positioning errors. Other techniques, such as Time of Arrival (ToA) and Time Difference of Arrival (TDoA), offered high accuracy in estimating the geolocations of sensor nodes by measuring signal propagation times between transmitters and receivers; however, these techniques require strict synchronization, which can be challenging to achieve in practical applications [6].

In the context of low-power and long-range communication, LoRaWAN has gained prominence, particularly in IoT applications where its potential for both indoor and outdoor localization has been explored. Recent studies have shown that by leveraging variations in RSSI and SNR, and integrating ML, LoRaWAN-based systems can achieve high localization accuracy, with some achieving up to 99% accuracy in indoor settings [6]. Despite the inherent challenges of indoor localization due to the need for multiple gateways, LoRaWAN has been shown to outperform other wireless technologies, such as WiFi and Bluetooth Low Energy (BLE), in specific scenarios, achieving mean localization errors as low as 2.7 meters [11].

Utilizing ML techniques in localization tasks has further enhanced its robustness and accuracy. For instance, in [6], the authors used RSSI-based approaches for indoor localization and evaluated the applicability of multiple ML models for LoRa mesh networks in GPS-denied environments. The proposed approach leverages various networking features such as Signal-to-Noise Ratio (SNR) and Time Difference of Arrival (TDoA) to enhance localization accuracy. Furthermore, emphasizing hyperparameter tuning to optimize model performance has provided an in-depth exploration of the trade-offs between model complexity, accuracy, and power efficiency.

III. Design and Implementation

A. Physical Implementation

As shown in Figure 1, the experimental setup involves Lora EDs equipped with an ESP32 microcontroller, an RFM95 LoRa chip, and a 6mm GPS module. The ESP32 microcontroller, with its dual-core processor, offers an excellent balance between performance and power consumption, supporting intensive tasks that require WiFi and Bluetooth connectivity. The RFM95 LoRa chip facilitates long-range communication and is particularly valued for its low power consumption, which is crucial for applications requiring extended battery life.

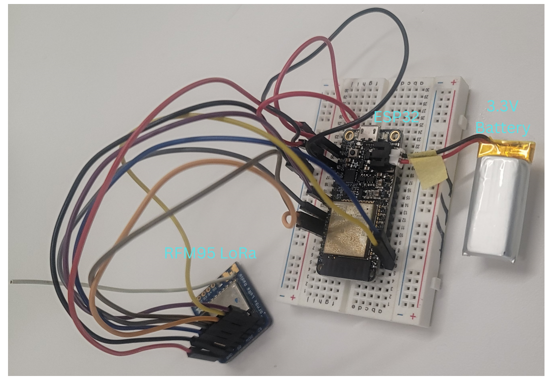


Fig. 1. The Hardware Design of a LoRa End Device.

Including a GPS module in traditional GPS-dependent systems typically escalates the power requirements of the location-tracking device. In our experiments, the 6mm GPS modules continuously receive and process signals from satellites, leading to significant battery drain, particularly in applications necessitating continuous location updates. By excluding the GPS module from our tracking system, we substantially reduce the power consumption of each ED node.

To quantify the energy savings, it's essential to understand the power consumption profiles of the different hardware components. (i) `GPS Module`: It consumes between 20 to 30mA during active tracking. Continuous operation of the GPS thus results in considerable power usage over extended periods. (ii) `LoRa RFM95 and ESP32 modules`: Both components consume much less power than the 6mm GPS module. The ESP32 can use as little as 10μA in deep sleep mode and about 150μA during active transmission with the LoRa network. The duty cycle of LoRa transmission is usually low, as data is transmitted intermittently, reducing the average power consumption.

B. Dataset Collection

The dataset comprises the geolocations of LoRa EDs data at a specific location during the experiment. As shown in Figure 2, the data collection setup consists of three static anchor nodes, represented by blue markers, and one mobile node, represented by a red marker. The anchor nodes are strategically placed within a defined geographic area to provide reliable reference points for the location tracking system, forming a unilateral triangle. The mobile node moves within the triangle's area, collecting data on signal strength indicators such as RSSI, SNR, and TDoA from the anchor nodes, as well as its GPS coordinates. This data is then used to train the ML models, which will predict the geographic coordinates of the mobile node.

The collected dataset comprises 60,000 entries and occupies approximately 5.5 MB of memory. Table I shows a sample subset of the collected dataset, where each tuple has the following attributes: (i) `Node ID`: A unique identifier for each node. (ii) `Latitude and Longitude`: Geographic coordinates of the node's location. (iii) `RSSI`

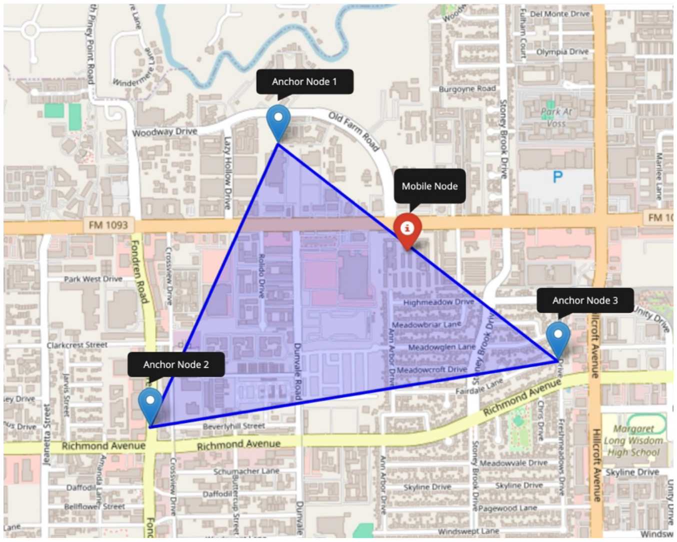


Fig. 2. Data Collection Experiment: Static Anchor Nodes (Blue) and Mobile Node (Red)

1, RSSI 2, RSSI 3: Received signal strength indication values from three different sources. (iv) SNR 1, SNR 2, SNR 3: Signal-to-Noise Ratio values from three different sources. (v) TDoA 1, TDoA 2, TDoA 3: Time Difference of Arrival values from three different sources.

C. Dataset Preprocessing

We conducted several data preprocessing steps before training the ML models, including data cleaning, formatting, feature selection, handling missing attributes, and duplication removal.

1) Cleaning and Formatting: : The initial preprocessing stage involved removing outliers, normalizing signal strength values, and converting data into a consistent format to ensure compatibility across different data sources. Specifically, signal values were filtered to remove extraordinarily high or low readings likely caused by environmental noise or hardware anomalies. Additionally, all data entries were formatted consistently to ensure proper alignment during the training phase.

2) Feature Selection: : The features selected for training the ML models included RSSI, SNR, and TDoA. These features were selected based on their direct relevance to signal strength and time of arrival, which are critical for determining each node's location without using GPS.

3) Handling Missing Attributes: : The dataset was meticulously checked for missing values to ensure completeness and reliability.

4) Duplication Removal: : Duplicate entries in the dataset were identified and removed. This step was essential to maintain the data integrity and prevent redundancy in the model training phase.

D. ML Model Training

The training phase involves preparing the ML models to accurately predict outcomes based on input features, including the RSSI, SNR, and TDoA. In this work, we employed the supervised learning approach, where four models are trained using labeled data. This process allows each model to learn the relationship between input features and their corresponding target labels, enabling predictions on fresh, unseen data samples.

1) **Linear Regression** is utilized to model the linear relationship between dependent and independent variables. It serves as a baseline model to establish an initial understanding of the linear associations within the data. This model is beneficial for understanding the direct impact of each predictor variable (e.g., RSSI, SNR, and TDoA) on the location coordinates.
2) **Random Forest Regression** offers advantages in handling overfitting and providing higher accuracy by building multiple decision trees and merging their outputs. It is robust against noise and capable of modeling complex, nonlinear relationships between the features and the target location variables. This makes it suitable for datasets with complex interaction effects and high dimensionality, like the one used in this work.
3) **K-Neighbors Regressor** is selected for its ability to predict outcomes based on localized patterns. By averaging the outputs of the k nearest data points in the feature space, this model can effectively capture subtle nuances in how signal strengths and timing differences affect the location estimation. It's beneficial when the geographical distribution of data points affects the outcome, as in urban or varied terrain settings.
4) **Decision Tree Regressor** provides an interpretable model structure that can easily depict decisions, which is vital for understanding how different signal features contribute to location predictions. Although susceptible to overfitting in noisy datasets, when appropriately tuned, it offers profound insights into the causal relationships within the data.

Upon training the models, we evaluated them to select the best deployment model based on performance metrics and suitability to task requirements. This process involves a detailed evaluation of each model's accuracy, robustness, and computational efficiency. The best model is selected based on its ability to provide the most accurate and reliable location predictions with reasonable computational demands.

We performed feature engineering to enhance the predictive performance of our ML models, which aims to predict latitude and longitude coordinates accurately. Feature engineering involves transforming raw data into informative features that can better represent the underlying patterns in the dataset.

Our input features (X) are derived from the dataset, excluding the 'Node ID', 'Latitude', and 'Longitude' attributes, as they serve as identifiers and target variables, respectively. These features represent various aspects of the environment or context from which we aim to predict latitude and longitude coordinates. We convert the RSSI, SNR, and TDoA attributes into a NumPy array (X) to facilitate model training and

TABLE I
A SAMPLE SUBSET OF THE COLLECTED DATASET

Node ID	Latitude	Longitude	RSSI 1	RSSI 2	RSSI 3	SNR 1	SNR 2	SNR 3	TDoA 1	TDoA 2	TDoA 3
1	30.095533	-95.990582	-67	-53	-62	12	16	26	0.402285	0.063060	0.774908
2	30.090537	-95.992485	-63	-62	-70	24	22	29	0.020667	0.321181	0.017177
3	30.102261	-95.986619	-55	-68	-52	12	20	17	0.864311	0.488032	0.704004
4	30.087665	-95.993680	-52	-59	-54	14	10	18	0.423811	0.844232	0.760452

evaluation. Our target variables (Y) consist of latitude and longitude coordinates that we seek to predict using X features.

We split the dataset into training and testing subsets to evaluate our model's performance. We allocated 70% of the data for training (`X_train`, `y_train`) and 30% for testing (`X_test`, `y_test`). The criteria for model selection include: (i) `Accuracy`: Measured by how closely the predicted location coordinates align with the true data points. (ii) `Robustness`: The model's ability to handle different variations in data and maintain performance across various scenarios. (iii) `Efficiency`: Considering the model's computational requirements and its feasibility for real-time application, given the constraints of device power and processing capabilities.

E. Evaluation Metrics

The cross-validation results for each model are evaluated to ensure its performance is consistent across different subsets of the data. Also, we performed a comparative analysis of various performance metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared values among the models. Furthermore, we considered the model's complexity and scalability potential for practical deployment within the LoRaWAN infrastructure.

The MAE measures the average absolute difference between the predicted latitude and longitude coordinates and the actual coordinates. A lower MAE indicates better accuracy. MAE is calculated as the average of the absolute differences between the predicted values ($\hat{y}_i$) and the actual values (y_i) for each data point (i), as follows.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{1}$$

where n is the number of data points, $\hat{y}_i$ is the predicted value for the i^{th} data point, and y_i is the true value for the i^{th} data point.

MSE measures the average squared difference between the predicted values ($\hat{y}_i$) and the actual values (y_i) for each data point (i), as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{2}$$

where n is the number of data points, $\hat{y}_i$ is the predicted value for the i^{th} data point, and y_i is the actual value for the i^{th} data point.

The RMSE is the square root of the Mean Squared Error (MSE) that measures the typical error in the predicted coordinates. Meanwhile, the Coefficient of Determination (R-squared) value measures the proportion of variance in the true coordinates, which the model explains. A higher R-squared value indicates better goodness of fit.

F. Hyperparameter Tuning

This section describes the process of hyperparameter tuning for various regression models to improve their performance.

1) Linear Regression: The linear regression model underwent minimal hyperparameter tuning because it inherently has fewer parameters to adjust. Typically, the tuning process for linear regression focuses on regularization parameters such as $L2$ (Ridge) and $L1$ (Lasso).

2) Random Forest Regressor: Hyperparameter tuning significantly improved the performance of the Random Forest model. The tuned hyperparameters included setting the number of trees (`n_estimators`) to 100, maximum depth (`max_depth`) to 15, the minimum number of samples required to split an internal node (`min_samples_split`) to 5, and the minimum samples of leaf nodes to 2. These adjustments led to a marked decrease in MAE, MSE, and RMSE and an improved R-squared value, indicating a better fit for the dataset.

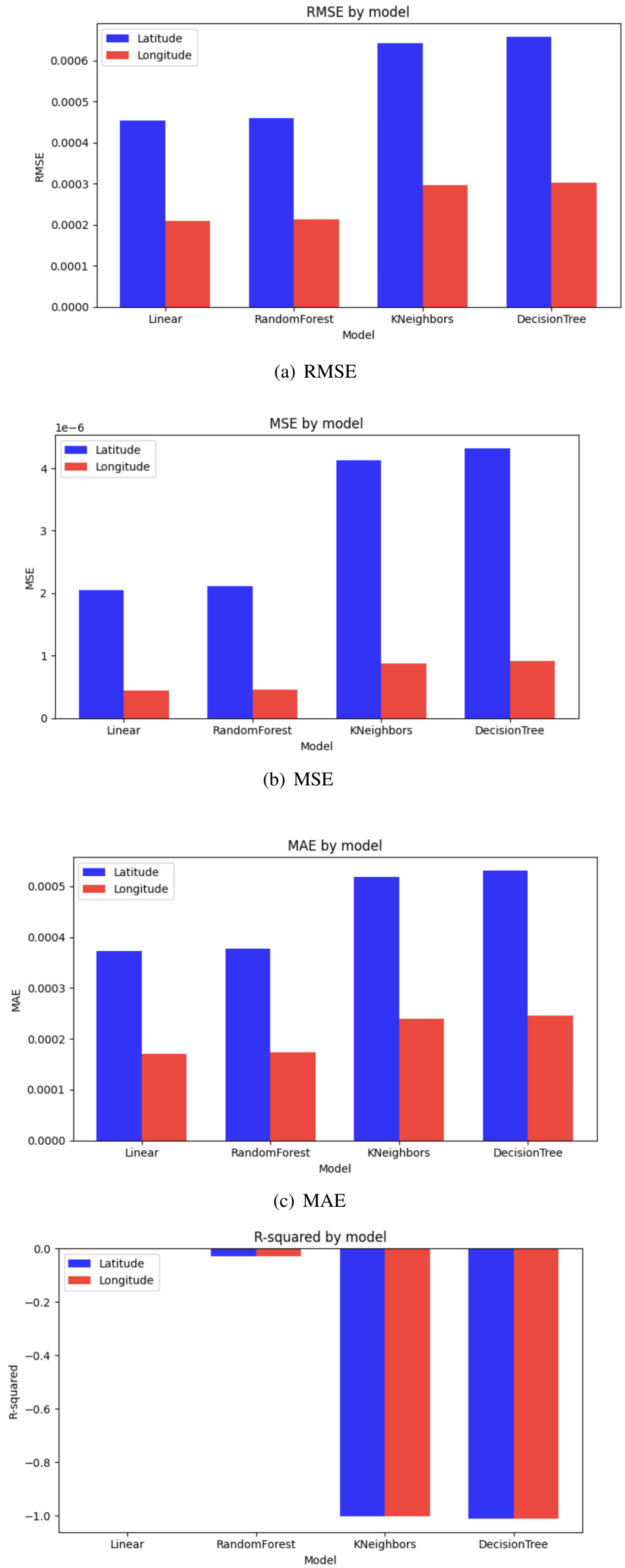
3) K-Neighbors Regressor: For the K-neighbors regressor, we adjusted the number of neighbors (`n_neighbors`) to 5, and the weighting function used (`weights`) to 'distance'.

4) Decision Tree Regressor: For the Decision Tree Regressor, we fine tuned several parameters such as the maximum depth (`max_depth`) to 10, the criterion for splitting (`criterion`) to 'friedman_mse', the minimum number of samples per leaf (`min_samples_leaf`) to 4, the minimum number of samples per split (`min_samples_split`) to 10.

IV. EXPERIMENTAL EVALUATION

The evaluation results highlight the trade-offs between each model's complexity and its performance in predictive modeling. Various evaluation metrics, including MAE, MSE, RMSE, and R-squared, are considered for predicting the latitude and longitude values. These metrics provide insights into the performance of each ML model in terms of prediction accuracy, error magnitude, and explained variance.

Figure 3 compares the performance of four ML models using the RMSE, MSE, MAE, and R-squared evaluation metrics. We found that the Linear Regression model achieved a substantial decrease in MAE, MSE, and RMSE values, indicating higher prediction accuracy and smaller error magnitudes. However, the R-squared values remain near zero, suggesting limited explanatory power.



(a) RMSE

(b) MSE

(c) MAE

(d) RSQUARED

Fig. 3. Comparing the Performance of ML models using the RMSE, MSE, MAE and R-Squared Metrics

Table II compares prediction metrics, including MAE, MSE, RMSE, and R-squared values, for various machine learning models used in latitude and longitude prediction tasks. The Random Forest model demonstrated superior performance for the latitude prediction with the lowest MAE and R-squared values, indicating a marginally better fit than the other models. Similarly, the Random Forest model achieved the lowest MAE and R-squared values in longitude prediction, suggesting its effectiveness in both tasks. While the Linear Regression model showed competitive results regarding RMSE, its R-squared values were notably lower, indicating potential limitations in capturing the underlying data patterns.

The evaluation results show that simpler models like linear regression can provide competitive performance for geographic coordinate prediction tasks. Despite the higher computational complexity and challenges in interpretability, Random Forest's ability to handle complex, non-linear patterns makes them the best choice for maximizing prediction accuracy and robustness in our geographic coordinate prediction task, as evidenced by their lower MAE, MSE, and RMSE values and improved R-squared scores. Conversely, the K-Neighbors and Decision Tree model may not be suitable for the location tracking task due to their higher errors and poor performance compared to Linear Regression and Random Forests models.

Table III compares the distance accuracy between the ground truth latitude and longitude values and the predicted ones by the four ML models before and after hyperparameter tuning. The table shows that Random Forest outperformed the other three models by achieving a high distance accuracy of less than one meter after hyperparameter tuning. It achieved superior distance accuracy and explanatory power by capturing non-linear relationships and interactions between features, which other models might miss. With its ensemble approach, the Random Forest model reduces the risk of overfitting and improves generalization, which is essential for our dataset characterized by varying signal strengths and time differences of arrival.

While K-Neighbors and Decision Tree Regressor models also attempted to model non-linear patterns, they exhibited higher errors and poorer performance compared to Random Forest and Linear Regression. This can be attributed to the sensitivity of the K-Neighbors Regressor to the choice of neighbors and distance metrics, which might not effectively capture the underlying patterns in the data. Similarly, the Decision Tree Regressor can suffer from overfitting, where the model becomes too complex and tailored to the training data, failing to generalize well to new, unseen data. Despite these models' capability to handle non-linearity, their performance is often less robust compared to Random Forest, which benefits from aggregating multiple decision trees to mitigate overfitting and enhance predictive accuracy.

Despite using existing ML models, this work is novel in applying them to GPS-denied environments using LoRa mesh networks. Integrating LoRaWAN with ML to solve real-world challenges and in-depth hyperparameter tuning adds valuable technical contributions. The small-scale experimental setup

TABLE II
EVALUATING THE ML IN PREDICTING THE LATITUDE AND LONGITUDE VALUES USING VARIOUS METRICS

Model	Latitude				Longitude			
	MAE	MSE	RMSE	R-squared	MAE	MSE	RMSE	R-squared
Linear Regression	1.50e-05	2.054809e-11	0.000004533	4.592e-08	1.25e-05	4.380649e-14	0.0000002093	8.231e-1
Random Forest	7.00e-06	2.113241e-11	0.000004597	0.00028379	6.00e-06	4.502884e-14	0.0000002122	0.000028044
KNeighbors Regressor	2.00e-05	4.124208e-11	0.000006422	0.000007095	1.75e-05	8.785296e-14	0.0000002964	0.0000005931
Decision Tree	2.50e-05	4.315176e-11	0.000006569	0.000100268	2.25e-05	9.174841e-14	0.0000003029	0.0000094895

TABLE III
DISTANCE ACCURACY (IN METERS) OF THE ML MODELS BEFORE AND AFTER HYPERPARAMETER TUNING

Model	Before Tuning	After Tuning
Linear Regression	6.72	1.98
Random Forest	3.25	0.93
K-Neighbors Regressor	4.65	2.68
Decision Tree Regressor	5.45	3.33

demonstrates the feasibility of using LoRaWAN for location tracking without GPS, providing a proof of concept for future larger deployments. The controlled environment allowed for careful monitoring of variables and provided a solid foundation to validate the ML models, ensuring that the methodology is sound before pursuing larger-scale applications. This work opens up new avenues for future research to explore hybrid ML models with traditional techniques to further improve prediction accuracy and model robustness for geographic coordinate prediction tasks in GPS-denied environments.

V. CONCLUSION

This paper presented a location tracking system leveraging LoRaWAN networks using various ML models. A significant advantage of the proposed system is its ability to function independently of GPS technology, thereby eliminating reliance on satellite signals, which may be unreliable or unavailable in specific contexts. Moreover, the non-dependence on GPS results in substantial energy savings, as the power-intensive nature of GPS can be circumvented, thereby extending the battery life of mobile devices used within the network. This experimental study evaluated various ML models for predicting geographic coordinates without GPS signals, including the Linear Regression, Random Forest, K-Neighbors Regressor, and Decision Tree Regressor models. Experimental results showed that the Random Forest model outperformed the other ML models, significantly improving accuracy and explanatory power. Although the Linear Regression model performed relatively well with low errors and good accuracy, it showed limitations in explaining variance in the data.

This work opens up new avenues for future research to explore hybrid ML models with traditional techniques to further improve prediction accuracy and model robustness for geographic coordinate prediction tasks in GPS-denied environments. Additionally, it is crucial to explore the environmental context, such as physical obstacles, terrain characteristics, and weather conditions, which can significantly affect the performance of ML models in real-world deployments. A detailed description of these factors will help understand the limitations and strengths of the proposed system in different scenarios and enhance the replicability of the experiments.

In addition, we plan to assess other traditional statistical or geometric estimation methods to estimate the longitude and latitude attributes. These methods may achieve results comparable to the ML approach, with reduced complexity and computational requirements.

ACKNOWLEDGMENT

This research work is supported in part by NSF under grants #2219611, 1910868, 2200377.

REFERENCES

[1] S. Bagwari, A. Roy, A. Gehlot, R. Singh, N. Priyadarshi, and B. Khan, "Lora based metrics evaluation for real-time landslide monitoring on iot platform," *IEEE Access*, vol. 10, pp. 46 392–46 407, 2022.

[2] M. Jouhari, N. Saeed, M.-S. Alouini, and E. M. Amhoud, "A survey on scalable lorawan for massive iot: Recent advances, potentials, and challenges," *IEEE Communications Surveys Tutorials*, vol. 25, no. 3, pp. 1841–1876, 2023.

[3] S. T. Ahmed and A. Annamalai, "On private server implementations and data visualization for lorawan," in *2023 IEEE 13th Symposium on Computer Applications Industrial Electronics (ISCAIE)*, 2023, pp. 342–347.

[4] D. Buddhi and A. Joshi, "Retracted: Tracking military soldiers location and monitoring health using machine learning and lora model," in *2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon)*, 2022, pp. 1–6.

[5] T. T. An, D. Q. Hien, T. T. Son, and D. H. Ai, "A trilateration-based positioning model for the real-time 2d map wireless sensor networks," in *2023 International Conference on Advanced Technologies for Communications (ATC)*. IEEE, 2023, pp. 399–403.

[6] T. Perković, L. D. Rodic, J. Šabić, and P. Šolić, "Machine learning approach towards lorawan indoor localization," *Electronics*, vol. 12, no. 2, p. 457, 2023.

[7] S. T. Ahmed and A. Annamalai, "Improving geo-location performance of lora with adaptive spreading factor," in *2023 IEEE 13th Symposium on Computer Applications Industrial Electronics (ISCAIE)*, 2023, pp. 386–391.

[8] S. T. Ahmed, A. A. Ahmed, A. Annamalai, and M. F. Chouikha, "A scalable and energy-efficient lorawan-based geofencing system for remote monitoring of vulnerable communities," *IEEE Access*, vol. 12, pp. 48 540–48 554, 2024.

[9] R. Berto, P. Napoletano, and M. Savi, "A lora-based mesh network for peer-to-peer long-range communication," *Sensors*, vol. 21, no. 13, 2021.

[10] S. T. Ahmed, A. A. Ahmed, A. Annamalai, and M. Chouikha, "Efficient rebroadcast location-unaware protocol for lorawan mesh networks in the iot domain," in *2024 IEEE 14th Symposium on Computer Applications Industrial Electronics (ISCAIE)*, 2024, pp. 104–110.

[11] B. C. Fargas and M. N. Petersen, "Gps-free geolocation using lora in low-power wans," *Journal of Communication Networks*, vol. 56, no. 4, pp. 1024–1031, 2023.

[12] F. Carrino, A. Janka, O. Abou Khaled, and E. Mugellini, "Loraloc: Machine learning-based fingerprinting for outdoor geolocation using lora," in *2019 6th Swiss Conference on Data Science (SDS)*, 2019, pp. 82–86.