Fingerprinting-based Indoor and Outdoor Localization with LoRa and Deep Learning

†Jait Purohit, †Xuyu Wang, ‡Shiwen Mao, †Xiaoyan Sun, and ‡Chao Yang
†Department of Computer Science, California State University, Sacramento, CA 95819, USA
‡Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849-5201, USA
Email:jaitpurohit@csus.edu, xuyu.wang@csus.edu, smao@ieee.org, xiaoyan.sun@csus.edu, czy0017@tigermail.auburn.edu

***Abstract*—This paper aims at predicting accurate outdoor and indoor locations using deep neural networks, for the data collected using the Long-Range Wide-Area Network (LoRaWAN) communication protocol. First, we propose an interpolation aided fingerprinting-based localization system architecture. We propose a deep autoencoder method to effectively deal with the large number of missing samples/outliers caused by the large size and wide coverage of LoRa networks. We also leverage three different deep learning models, i.e., the Artificial Neural Network (ANN), Long Short-Term Memory (LSTM), and the Convolutional Neural Network (CNN), for fingerprinting based location regression. The superior localization performance of the proposed system is validated by our experimental study using a publicly available outdoor dataset and an indoor LoRa testbed.**

I. Introduction

Due to the explosive increase in Internet of Things (IoT) applications and location-based services, where billions of devices and gateways are involved, wireless positioning techniques in Low Power Wide Area Network (LPWAN) become an important problem [1]–[4]. Currently, outdoor location information is mostly obtained with the global positioning system (GPS), which can achieve a localization accuracy of about 5 m in line-of-sight (LOS) conditions for civilian use [5]. However, GPS suffers from bad performance in outdoor, rich-scattering environments and urban canyons. Moreover, GPS receivers are not only power hungry, which greatly limits the battery lifetime, but also too costly to be integrated into many IoT devices [3]. Since GPS does not work indoors, where many location-based services are offered, it does not provide an integrated indoor/outdoor solution. Alternative outdoor localization methods are proposed, e.g., using long term evolution (LTE) with observed-time-difference-of-arrival (OTDOA) [6], massive multiple-input and multiple-output (MIMO) in sub-6 GHz [7], as well as mmWave [8]. However, these techniques usually have high power consumption, making them not suitable for IoT devices.

Fingerprinting based solutions with deep learning have been proposed recently, which are highly suitable for non-line-of-sight (NLOS) environments and can achieve better performance than traditional machine learning based schemes. This is because the received signal from NLOS can be used as features for location estimation with deep learning. Generally, fingerprinting-based localization requires a training phase to create a database of many location and data pairs, and a test phase to search for the most matched fingerprint for location estimation. Our previous work, termed DeepFi, is the first to apply deep learning for localization using WiFi channel state information (CSI) amplitude [9], [10]. Our other works apply an autoencoder to handle CSI phase difference and bi-modal CSI data [11], [12]. In addition, our recent work CiFi is the first to utilize a deep convolutional neural network (DCNN) for indoor localization using images constructed with CSI phase difference data between different subcarriers [13]. On the other hand, although there are several Long Range (LoRa)-based outdoor fingerprinting schemes [1], [2], they only use traditional machine learning methods, e.g., k-Nearest Neighbors (KNN) [14]–[16], Support Vector Regression (SVR) [17], and Bayesian methods [18], which are not very effective in exploiting LoRa signal data for high localization accuracy.

In this paper, we address the problem of fingerprinting based localization in complex indoor and outdoor environments. The goal is to design an efficient, indoor/outdoor integrated solution with the LoRa technology and deep learning techniques. This approach will provide a better location estimation solution in comparison to other machine learning-based approaches due to the higher learning capability of the deep learning models. First, we will introduce the background of the LoRa technique. Then, we present the system design for the LoRa-based localization system. We will also introduce different interpolation methods including linear, cubic, quadratic methods, and propose a deep autoencoder method to improve the interpolation performance with LoRa signals. Moreover, we incorporate three different deep learning models, including the Artificial Neural Network (ANN), Long Short-Term Memory (LSTM), and the Convolutional Neural Network (CNN), into the fingerprinting system and evaluate their performance. Finally, we use hyperparameter tuning for the deep learning networks to reduce their location errors. We evaluate the localization performance of the proposed schemes, and compare their performance with several baseline schemes, using an open-source LoRa dataset for outdoor experiments and a LoRa testbed for indoor experiments. Our experimental study demonstrates that the deep learning methods can achieve a superior localization performance than the traditional machine learning based schemes.

In the following, the preliminaries are introduced in Section II. We present the system design and the performance study in Section III and our performance study in Section IV. Section V summaries this paper.

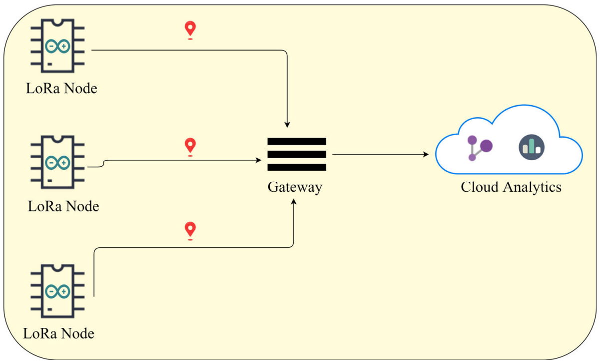

Fig. 1. Sensor-gateway interaction over the LPWAN technology.

II. PRELIMINARIES

A. Long-Range Wide-Area Network (LoRaWAN)

The fourth industrial revolution is heavily dependent on IoT networks where billions of devices are interconnected over the Internet. The challenges related to these devices include power efficiency, cost-constraints, and the ability to communicate reliably over long ranges. Long-Range Wide-Area Network (LoRaWAN) is a relatively new protocol for LPWAN, which provides solutions to address these issues.

LoRaWAN is a widely adopted proprietary technology for long range communications, where the chirp spread spectrum (CSS) modulation technique is employed for LoRa [1]. LoRa operates in different frequency bands at different geolocations. The Europe region uses the frequency band from 863 MHz to 870 MHz, the US region uses the band from 902 MHz to 928 MHz, while China operates between 779 MHz to 787 MHz [1]. In addition, LoRa symbols can be encoded using a number of chirps, which spreads the signal over various channels. This technique helps to reduce interference with other signals. The Spreading Factor (SF) determines the number of chirps needed, which ranges from 7 to 12. An SF value closer to 12 means longer ranges, which is achieved at the expense of low data rates in comparison to a lower spreading factor. The relation between SF and signal range has a direct effect on the received signal strength indicator (RSSI) and distance mapping [1], [14].

As shown in Fig. 1, a LoRa node (or, sensor) communicates with a single or multiple channel gateways over a single hop, resulting in a TCP/IP uplink (sensor to gateway) to the LoRaWAN cloud server like the Things Network [19]. The network server is mainly responsible for passing messages between edge sensors and applications. Any type of communications in this network architecture will be encrypted twice, i.e., once using the network session key and again using the application session key. Microservices to handle gateway traffic are written in the publisher and subscriber model using the Apache ActiveMQ-MQTT broker [1].

End-nodes are physical hardware sensor devices that contain sensing capabilities and computing power up to some extent. Gateways, also known access-points, are used to pick up all message payloads from edge devices. These payloads or radio frequency (RF) packets are converted to IP packets (in arrays of bits) over the network server and are further sent through traditional IP networks to the application server. The application server is the place where the actual IoT application is residing and handles the data collected from edge devices. Application servers can run mostly on the cloud to perform advanced analytics, data preprocessing, and create RESTful web-services [1].

Compared to other communication technologies like Sigfox and NB-IoT, LoRa helps to enable localization using Time Difference of Arrival (TDoA) [20], where accurate synchronization amongst receiving gateways is required. Fingerprinting based localization methods using LoRa RSSI values have been proposed, which have a low cost and are easy to deploy and operate. In fact, fingerprinting based localization with LoRa signals using traditional machine learning techniques does not help to achieve high outdoor localization accuracy. This is because of the large amount of data streams from an enormous number of LoRa nodes, which are hard to be fully exploited by traditional machine learning menthods.

B. Problem Statement

In this paper, we aim to address fingerprinting based localization problems in complex indoor and outdoor environments. The challenges include the non-linearity, multipath, and obstacles. We aim to design an efficient solution with LoRa to accurately estimate location using deep learning based methods. The deep learning techniques used are supervised learning based on the LoRa data, which can better handle non-linear Gaussian noise than other traditional approaches, such as KNN, and are more accurate as they can adjust neural weights and the number of hidden layers with hyperparameter tuning to reduce the mean location error.

III. SYSTEM DESIGN

Fingerprinting-based localization requires a specific setup in terms of hardware and software, which can be pre-configured in the cloud [19]. This approach is divided into two phases: an offline (training) phase and an online (testing) phase. In the offline phase, RSSI values are collected for each predefined location. The data samples are stored in a database.

The online phase is a testing phase where RSSI values are known and the testing locations (i.e., the latitude and longitude) in indoor or outdoor environments are unknown. Using deep-learning techniques can estimate these testing locations and reduce the location error. The training of the model is done using TensorFlow framework [21], Keras, and Scikit-Learn libraries. Additionally, Google Colab has been used as a free cloud service to train these models as it provides hardware accelerators such as graphics processing unit (GPU) and tensor processing unit (TPU).

A. System Architecture

The architecture of LoRa-based localization is shown in Fig. 2, illustrating the high-level architectural components. A LoRa sensor node sends data payloads to multiple Gateways.

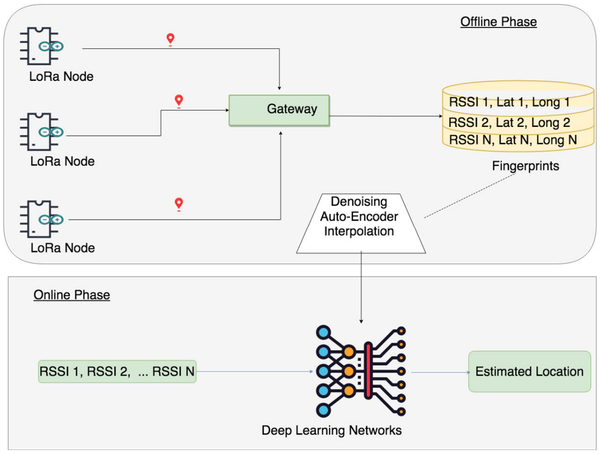

Fig. 2. System architecture of LoRa based localization.

The RSSI of the LoRa signal is an important indicator to measure the signal quality at base-stations or gateways. The other information, such as SF and horizontal dilution of precision (HDOP), are also collected at the Gateway endpoint and stored in the form of a time-series in a database as a part of the training (offline) phase. At each location, several instantaneous RSSI readings are collected as a part of the offline phase. The instability and fluctuations of RSSI values are mostly due to the multipath effect.

Deep learning based LoRa localization helps to address the multipath issues. Deep networks e.g., LSTM, can handle Gaussian noise better than traditional machine learning algorithms. LSTM handles sequential data efficiently for fingerprinting based localization, as it compares the LoRa node's current location with the node's previous location in the same trajectory. Moreover, as the size of sensor data increases, it becomes difficult to train the basic machine learning models. Deep learning models are more effective in handling issues related to spatial ambiguity and RSSI instability. In this paper, we consider three different deep learning models (i.e., ANN, LSTM, and CNN) as regression methods for LoRa-based indoor and outdoor localization.

Before implementing a deep learning model, data preprocessing is executed to remove outliers using interpolation techniques. In particular, denoising auto-encoder network, one of the interpolation methods, can be leveraged as a part of the training phase to extract information of the outliers. Auto-encoders are a type of neural networks used to extract encoded data information in an unsupervised manner and decode true features of the dataset. We will discuss the detailed implementations of interpolation and the deep learning model in the following.

B. Data Preprocessing with Interpolation

From received LoRa singals, useful information, such as RSSI, ID, timestamp, spreading factor (SF), HDOP, and geolocation co-ordinates, can be extracted. Each base-station can have different channel conditions, and thus there could be many missing data samples (and outliers) in the process, especially in outdoor environments, which can greatly degrade the localization accuracy. Therefore, such missing data samples should be interpolated using the measured dataset, using interpolation techniques such as linear, cubic, quadratic, and denoising auto-encoder.

Linear interpolation used here is a method of fitting the curve using linear polynomials to estimate new data points within some discrete range of known data points. Similarly, we implement cubic and quadratic interpolation as well, where quadratic interpolation uses second-order polynomials and cubic uses third-order polynomials. Therefore, base-stations with missing values interpolate the missing values using forward or backward pattern of known data. In addition, autoencoders have also been used as a part of deep learning neural networks, which are mainly used for feature extraction, data denoising and reconstruction [22]. In our problem, due to the large amount of LoRa nodes, many data samples are missing (could be as high as 50%). For such cases, denoising autoencoders can solve the problem by corrupting the missing and outlier data and converting them to null values.

In our implementation of the denoising autoencoder, we use a function to shuffle data around and learn more about the data by attempting to reconstruct it. This process of shuffling helps to learn the features within the noise and will allow us to classify the input values. While training the neural network, it generates a model and measures the distance between the benchmark that has been set and the model through a loss function. The training process attempts to reduce the loss function by resampling the shuffled inputs and re-constructing the data-value until it finds those inputs which are true to the actual value. Therefore, the entire process attempts to convert missing samples and outliers to null values and the autoencoder must then denoise or learn to reconstruct it back, minimizing the log-loss function.

We compare the performance of linear, cubic, and quadratic interpolation functions (implemented with python SciPy inbuilt library functions), and the denoising autoencoder. Table I presents the comparison of interpolation results of LoRa signals of 100 random points for each of the two base-stations. The original data is divided into 70% training and 30% testing datasets. After interpolation, the fingerprint map for each base-station has been generated as training data. The difference between the original RSSI value and interpolated RSSI value has been calculated for each test data. It can be seen that the denoising autoencoder outperforms the other three techniques in terms of lower average error and lower standard deviation from its interpolated data. We also compare our results with that from fingerprint maps and find the denoising auto-encoder perform well in extrapolated areas.

C. Deep Neural Networks

Deep neural networks are also a class of machine learning algorithms that relies on non-linear processing neurons for feature extraction. Deep neural networks require specific type of hardware accelerators such as GPU or TPU, random-access memory (RAM), physical memory, and storage depending on

TABLE I
COMPARISON OF INTERPOLATION METHODS

Gateway	***Interpolation Algorithm***	***Average***	***Standard Deviation***
BS-1	Linear	7.85	7.52
	Cubic	10.12	9.23
	Quadratic	12.47	10.63
	Denoising Autoencoder	**6.52**	**3.79**
BS-2	Linear	6.97	7.17
	Cubic	9.24	8.47
	Quadratic	10.47	9.37
	Denoising Autoencoder	**5.62**	**3.47**

the complexity of the problem at hand. They have multiple hidden layers that help to model complicated functions. Non-linear data features are usually handled by deep neural networks and are useful in big data use cases. In this paper, we mainly use the following three types of deep neural networks, i.e., ANN, LSTM, and CNN, which are discussed below.

1) Artificial Neural Networks: ANN belongs to the class of machine learning models, where the computation of each neuron takes place internally and the networks are used to have inter-connectivity amongst them. Neurons from the current layer receive input from previous layers for further computation along with weights adjusted. The connection between these interconnected neurons is identified by weights and learning parameters, which are updated during training to get a favorable output. This type of networks are feed-forward neural networks (no loop connections), where the first layer is an input layer, followed by hidden layers and an output layer. Data is transferred using hidden layers from input to the output layer. There are several parameters, such as activation function, optimizer, epoch, and batch-size, used to configure and train the ANN model. Specially, four dense layers for the ANN model in this paper are used for LoRa dataset, where two nodes (i.e., latitude and longitude) are exploited for the last dense layer.

2) Long Short-Term Memory: LSTM is used to deal with sequence prediction problems, which can remember patterns for a longer duration of time. Compared with recurrent neural network (RNN), LSTM avoids long-term dependency problems, resulting into better accuracy [23]. LSTM also has a different structure for the repeating module. The cell state has limited linear interactions through the whole chain. There is a gate-like structure that helps to add or remove information in the cell state. These gates allow information to flow through and are composed of a Sigmoid function as an activation function to perform pointwise multiplication operation.

In our implementation of LSTM, to optimize fingerprinting localization, we have created an x-array and a y-array matrix. These matrices are returned as numpy arrays while calling the sequence function. These sequence functions are set to size of 1, which helps provide results for the latitude and longitude for each LoRa node. The LSTM model is sequential in nature and has a linear stack of layers. This model can be passed with the input-shape argument to the first layer as well as the three dense layers.

Dropout is a regularization method where recurrent connections to LSTM and inputs are removed from activation and weight changes while training the network. Dropout is used here to avoid overfitting and to improve model performance in case of indoor localization as we have very few data-points.

3) Convolutional Neural Network: CNN is a class of deep-learning algorithms which are usually used to deal with computer vision problems, such as image recognition, digit recognition, or object recognition [24]. The other concepts in CNN are related to padding, which helps to preserve the dimensions of input in the output label. The pooling layer is similar to the convolutional layer, which is helpful to reduce the spatial dimensions of the network by creating feature maps. It provides an approach of down-sampling. Max-pooling and average pooling are different types of pooling layers in the CNN architecture. In our implementation, we have used CNN as a regression model to estimate locations, where two convolutional layers and two dense layers are used for the CNN model.

D. Hyperparameter Tuning

Hyperparameter tuning is a part of model optimization to minimize the testing error [25]. Choosing the correct number and diversity of these parameters is dependent on each classifier and can vary accordingly. We have implemented hyperparameter tuning in the neural network models to minimize localization error by using different permutations and combinations of optimal parameters, such as batch-size, learning-rate, optimizer, activation function, and hidden layers.

IV. EXPERIMENTAL STUDY

A. Outdoor Dataset

The outdoor experiment is carried out using a publicly available LoRaWAN dataset [14]. Fig. 3 shows the data collected over a period of 3-4 months from Antwerp, Belgium. The goal of their approach was to create a benchmark to evaluate localization in outdoor environments using the kNN approach. The dataset consists of 123,529 LoRaWAN messages received at 68 base-stations, which are the gateways to transfer data from LoRa devices to the application layer. The LoRa nodes are spatially scattered over a larger radius in the city area and thus the location estimation has a relatively large error.

B. Indoor Experimental Setup

Indoor localization was implemented using a specific setup using Dragino LoRa gateways and a sensor node sending data payloads at different training and testing locations. The hardware required for this experiment is LoRa Dragino Kit, which includes Arduino UNO, LoRa Gateway, different sensors, LoRa GPS shield, and required cables for connections. The software part of LoRa setup has been done in C, while model training and evaluation have been done in python. The hardware setup is shown in Fig. 4.

We choose Riverside Hall, 3rd Floor, at Sacramento State to carry out our experiments. The data-collection strategy

Fig. 3. Map of the outdoor localization dataset from Antwerp, Belgium [14].

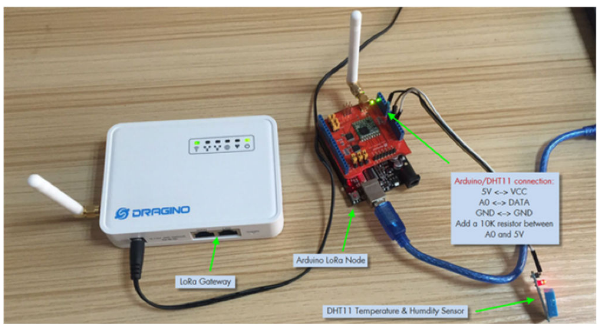

Fig. 4. The LoRa node hardware configuration.

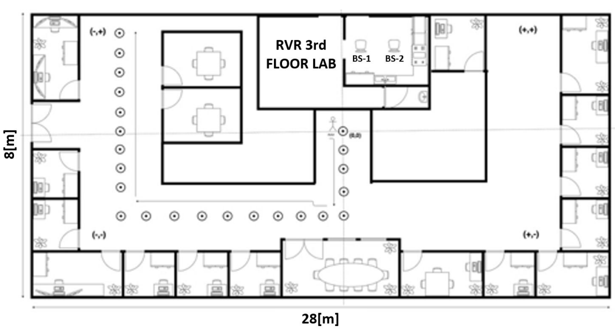

Fig. 5. The floor map of Riverside Hall, 3rd Floor at Sacramento State for indoor localization.

has been carried out, i.e., dividing the floor into a (X, Y) coordinate system, so as to collect 2D data in all directions of the floor as a part of offline phase. As shown in Fig. 5, Dragino LoRa LG01 gateways, i.e., the base-stations, have been set up in the RVR 3rd floor lab, at fixed locations. The black arrows in Fig. 5 indicate the user's walking trajectory; the DHT11 sensor sends data to these gateways from different locations. The horizontal corridor is 28 m long, while vertical corridor is 8 m long. The training and testing data are randomly collected within 1-3 m of difference.

The RSSI value is the most important indicator of the received LoRa signal. The values are measured in dBm and can take values from 0 dBm (excellent strength) to -120 dBm (extremely poor). The other features that we can measure

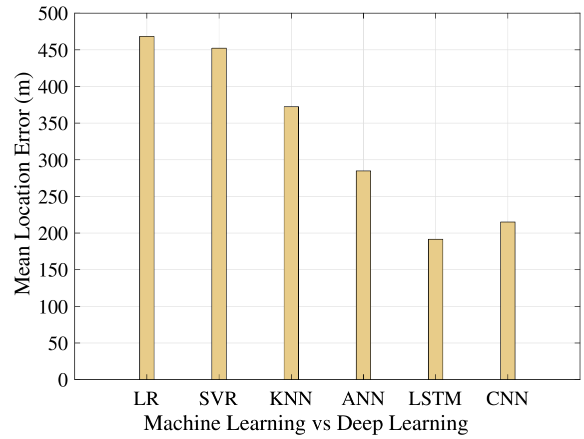

Fig. 6. Outdoor localization results (mean location errors).

in the process are the SF, which is the duration of the packets received, and HDOP, which measures the GPS signal quality using the satellite configuration. Note that HDOP is not used for the indoor localization experiments. Latitude and Longitude are denoted in the (X, Y) coordinate format.

C. Experimental Results

Fig. 6 presents the LoRa based outdoor localization rebsults from deep learning models and traditional methods. We can see that the deep-learning models outperforms the basic machine learning models, i.e., KNN [14], SVR, and Linear regression (LR). Moreover, LSTM achieves the best mean location error of 191.52726 m using 64 neurons, ReLu as activation function, and Adam as an optimizer, with a batch-size of 512, epochs as 10, and dropout as 0.1 to avoid over-fitting. The mean squared error is used as the loss function. For the LSTM model, 'sequence-size' is kept to 1. In many cases, the training of the model converges very fast. The validation split was 0.3, and the batch size was tweaked to remove biases and variance in the model. The training loss was 0.1360 and validation loss was 0.0911 when the model stops its training after the 5th epoch.

The mean error achieved by various deep learning model configurations are presented in Table II. ANN and CNN also perform better than the basic machine learning models. The best ANN model achieves a mean error of 284.7837 m with a batch size of 256 and epoch size of 20. On the other hand, CNN with a batch size 64 and epoch size of 3, can achieve a mean error of 215.06 m. Although KNN achieves the best performance among the three traditional machine learning schemes (i.e., LR, SVR and KNN), its mean error is 372.37 m (the mean error reported in [14] using KNN and the same dataset is 398.4 m), which is much larger than that of the deep learning based approaches.

The indoor data collected using the LoRa testbed discussed above is split according to a 70%:30% ratio. Table III presents the indoor localization errors using different deep learning models with different configurations. We can see that all the deep learning models can achieve a mean location error under 2 m. In addition, we can see that ANN and LSTM performs

TABLE II
MEAN LOCATION ERRORS OF THE OUTDOOR EXPERIMENT

Models	***Epoch***	***Batch Size***	***Dropout***	***Mean Error (m)***
ANN-1	10	64	None	284.78475
ANN-2	20	256	None	284.78366
ANN-3	10	512	None	284.81696
LSTM-1	10	512	0.1	191.52726
LSTM-2	12	256	0.5	194.77348
CNN-1	3	64	0.3	215.06072
CNN-2	3	81	0.5	221.75332

TABLE III
MEAN LOCATION ERRORS OF THE INDOOR EXPERIMENT

Models	***Epoch***	***Batch Size***	***Dropout***	***Mean Error (m)***
ANN-1	10	256	None	1.324271
ANN-2	15	512	None	1.270332
ANN-3	20	256	None	1.286759
LSTM-1	10	512	0.1	1.409174
LSTM-2	10	81	0.5	1.348190
LSTM-3	20	256	0.3	1.799690
CNN-1	3	64	0.5	1.804363
CNN-2	3	81	0.5	1.886141
CNN-3	3	128	0.5	1.786397

better than CNN. Because we only use 2 base stations in the experiment, it is not easy to create high-dimensional image data to improve the accuracy using CNN based methods. The indoor experimental results demonstrate that LoRa signals with RSSI values can be effective for indoor localization.

V. CONCLUSION

In this paper, we presented deep learning based indoor and outdoor localization with LoRa. We presented the system design, including fingerprinting based system architecture, interpolation methods, and three deep learning models, i.e., ANN, LSTM, and CNN. Our experimental results showed that deep learning methods can achieve satisfactory localization accuracy using LoRa signals in both indoor and outdoor scenarios.

ACKNOWLEDGMENTS

This work is supported in part by the NSF under Grant ECCS-1923163 and the Wireless Engineering Research and Education Center at Auburn University, Auburn, AL, USA.

REFERENCES

[1] N. Sornin, M. Luis, T. Eirich, T. Kramp, and O. Hersent, "LoRaWAN©specification," 2015, LoRa Alliance. [Online]. Available: https://lora-alliance.org/resource-hub/lorawanr-specification-v11

[2] C. Gu, L. Jiang, and R. Tan, "LoRa-based localization: Opportunities and challenges," *arXiv preprint arXiv:1812.11481*, Jan. 2019. [Online]. Available: https://arxiv.org/abs/1812.11481

[3] H. Sallouha, A. Chiumento, S. Rajendran, and S. Pollin, "Localization in ultra narrow band IoT networks: Design guidelines and tradeoffs," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9375–9385, Dec. 2019.

[4] Y. Li, Y. Zhuang, X. Hu, Z. Gao, J. Hu, L. Chen, Z. He, L. Pei, K. Chen, M. Wang *et al.*, "Location-enabled IoT (LE-IoT): A survey of positioning techniques, error sources, and mitigation," *arXiv preprint arXiv:2004.03738*, Apr. 2020. [Online]. Available: https://arxiv.org/abs/2004.03738

[5] S. Sadowski and P. Spachos, "RSSI-based indoor localization with the Internet of Things," *IEEE Access J.*, vol. 6, no. 1, pp. 30 149–30 161, June 2018.

[6] S. Fischer, "Observed time difference of arrival (OTDOA) positioning in 3GPP LTE," June 2014, qualcomm White Paper. [Online]. Available: https://www.qualcomm.com/media/documents/files/otdoa-positioning-in-3gpp-lte.pdf

[7] M. Arnold, J. Hoydis, and S. ten Brink, "Novel massive MIMO channel sounding data applied to deep learning-based indoor positioning," in *Proc. 12th Int. ITG Conf. Systems, Commun. Coding*, Rostock, Germany, Feb. 2019, pp. 1–6.

[8] F. Lemic, J. Martin, C. Yarp, D. Chan, V. Handziski, R. Brodersen, G. Fettweis, A. Wolisz, and J. Wawrzynek, "Localization as a feature of mmWave communication," in *Proc. IWCMC'16*, Paphos, Cyprus, Sept 2016, pp. 1033–1038.

[9] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-based fingerprinting for indoor localization: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 763–776, Jan. 2017.

[10] X. Wang, L. Gao, S. Mao, and S. Pandey, "DeepFi: Deep learning for indoor fingerprinting using channel state information," in *Proc. IEEE WCNC 2015*, New Orlean, LA, Mar. 2015, pp. 1666–1671.

[11] X. Wang, L. Gao, and S. Mao, "CSI phase fingerprinting for indoor localization with a deep learning approach," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1113–1123, Dec. 2016.

[12] X. Wang, L. Gao, and S. Mao, "BiLoc: Bi-modality deep learning for indoor localization with 5GHz commodity Wi-Fi," *IEEE Access J.*, vol. 5, no. 1, pp. 4209–4220, Mar. 2017.

[13] X. Wang, X. Wang, and S. Mao, "Deep convolutional neural networks for indoor localization with CSI images," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 316–327, Jan./Mar. 2020.

[14] M. Aernouts, R. Berkvens, K. Van Vlaenderen, and M. Weyn, "Sigfox and LoRaWAN datasets for fingerprint localization in large urban and rural areas," *MDPI Data*, vol. 3, no. 2, p. 13, Apr. 2018.

[15] G. G. Anagnostopoulos and A. Kalousis, "A reproducible analysis of RSSI fingerprinting for outdoor localization using Sigfox: Preprocessing and hyperparameter tuning," in *2019 Int. Conf. Indoor Positioning Indoor Navigation (IPIN)*, Pisa, Italy, Sept./Oct. 2019, pp. 1–8.

[16] K.-H. Lam, C.-C. Cheung, and W.-C. Lee, "RSSI-based LoRa localization systems for large-scale indoor and outdoor environments," *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 11 778–11 791, Dec. 2019.

[17] F. Lemic, V. Handziski, M. Aernouts, T. Janssen, R. Berkvens, A. Wolisz, and J. Famaey, "Regression-based estimation of individual errors in fingerprinting localization," *IEEE Access J.*, vol. 7, pp. 33 652–33 664, Mar. 2019.

[18] W. Choi, Y.-S. Chang, Y. Jung, and J. Song, "Low-power LoRa signal-based outdoor positioning using fingerprint algorithm," *MDPI Int. J. Geo-Information*, vol. 7, no. 11, p. 440, Nov. 2018.

[19] N. Blenn and F. Kuipers, "LoRaWAN in the wild: Measurements from the things network," June 2017, arXiv preprint arXiv:1706.03086. [Online]. Available: https://arxiv.org/abs/1706.03086

[20] N. Podevijn, D. Plets, M. Aernouts, R. Berkvens, L. Martens, M. Weyn, and W. Joseph, "Experimental TDoA localisation in real public LoRa networks," in *Proc. IEEE 10th Int. Conf. Indoor Positioning and Indoor Navigation*, Pisa, Italy, Sept./Oct. 2019, pp. 211–218.

[21] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. USENIX OSDI'16*, Savannah, GA, Nov. 2016, pp. 265–283.

[22] B. K. Beaulieu-Jones and J. H. Moore, "Missing data imputation in the electronic health record using deeply learned autoencoders," in *Proc. 2017 Pacific Symp. Biocomput.*, Big Island, HI, Jan. 2017, pp. 207–218.

[23] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," in *Proc. 1999 Int. Conf. Artificial Neural Networks*, Edinburgh, UK, Sept. 1999, pp. 850–899.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. NIPS'12*, Lake Tahoe, NV, Dec. 2012, pp. 1097–1105.

[25] J. Rodriguez, "Understanding hyperparameters optimization in deep learning models: Concepts and tools," Aug. 2018. [Online]. Available: https://towardsdatascience.com/