Deep Learning based Indoor Positioning Approach Using Wi-Fi CSI/RSSI Fingerprints Technique

Marco Mühl and Wiem Fekih Hassen

Chair of Distributed Information Systems, University of Passau, Innstraße 41, 94032 Passau, Germany
Email: {firstname.lastname}@uni-passau.de

***Abstract*—Indoor Positioning Systems (IPSs) play a vital role in various applications, ranging from asset tracking to location-based services. With different approaches being explored in the last years, Wi-Fi-based IPSs utilizing Channel State Information (CSI) and Received Signal Strength Indicator (RSSI) have gained increased attention. This research aims to develop a Wi-Fi based indoor positioning system using CSI and RSSI measurements, specifically focusing on datasets collected at the University of Passau, since datasets used in related work are private. Additionally, after the acquired data is subjected to preprocessing and data cleaning techniques, the study explores the potential of Machine Learning (ML) techniques, including Support Vector Regression (SVR), Long Short-Term Memory (LSTM), and Convolutional Neural Networks (CNN), to enhance positioning accuracy. These models are trained and evaluated using appropriate performance metrics, including Mean Squared Error (MSE) and distance error. The experimental results, focusing on the prediction of vertical and horizontal coordinates within the laboratory room, demonstrate the effectiveness of the proposed system. For unseen RSSI data, the best distance error based on MSE achieved was 29.5 cm using SVR, while for unseen CSI Amplitude data, the lowest distance error based on MSE was 37.9 cm with a CNN approach. A comparison is conducted within the different methods. All tested models consistently achieve a distance error based on MSE of under 50 cm, proving the high quality of the collected dataset. Future research directions and areas for improvement are also suggested.**

***Keywords*—IPSs, CSI, RSSI, private dataset, Raspberry Pi, SVR, LSTM, CNN.**

I. Introduction

Smart Things and connected devices have become an integral part of people's daily lives as well as various industries, such as healthcare and manufacturing. One of the key areas that is driving this trend is location-based services. While for outdoor applications the Global Positioning System (GPS) is an established standard [1], Indoor Positioning Systems (IPSs) face different challenges [2].

Compared to outdoor positioning, some of these challenges are: Need for higher accuracy, while having higher levels of signal interference, need for low power consumption, and handling environmental changes [3]. Since the demand for IPSs is rising, different approaches to solving these problems have emerged. With some approaches utilizing Radio-Frequency Identification (RFID), Bluetooth, or Ultra-Wideband (UWB) the use of Wi-Fi-signals, which are available in almost every indoor environment, has become a good option to provide accurate indoor localization [4]. They are compatible with almost every mobile device, are low cost, and have wide signal coverage.

Various signal metrics, such as Channel State Information (CSI) and Received Signal Strength Indicator (RSSI) can be employed in Wi-Fi-based IPSs, as well as different positioning techniques, like proximity, multilateration, angulation, or fingerprinting, to determine the location of a device within an indoor environment [5], [6]. In recent years, the application of Machine Learning (ML) algorithms has exhibited considerable improvements in the performance of IPSs [7]. These algorithms have the potential to effectively analyze the complex patterns and relationships present in the collected data, enabling accurate localization and tracking.

Since no CSI dataset for indoor positioning used in related work are publicly available, one primary objective of this work is to create an extensive CSI and RSSI dataset for a laboratory at the University of Passau. Subsequently, a comparative analysis of different Deep Learning (DL) approaches, including Support Vector Regression (SVR), Long Short-Term Memory (LSTM), and Convolutional Neural Networks (CNN), will be conducted to evaluate their effectiveness in achieving accurate indoor localization.

The remainder of this paper is structured as follows: Section II starts by outlining the proposed approach of a Wi-Fi-based indoor positioning system. After that, the data collection steps and the processing techniques for both CSI and RSSI data are detailed. Section III presents the adopted ML algorithms, i.e., SVR, LSTM and CNN. Section IV describes a comparative analysis of the results. Section V concludes the paper and identifies future challenges.

II. Proposed Methodology

A. Overall System Architecture

This subsection presents the proposed system architecture and the methods employed in this work. Our system is composed of three main phases, as depicted in Figure 1:

- *Phase 1*: consists of creating a dataset of RSSI and CSI values based on the specific collection tool Nexmon on a Raspberry Pi.
- *Phase 2*: involves the application of multiple data processing methods to clean up the datasets and prepare the fingerprint dataset.
- *Phase 3*: details the different implemented ML algorithms and their results.

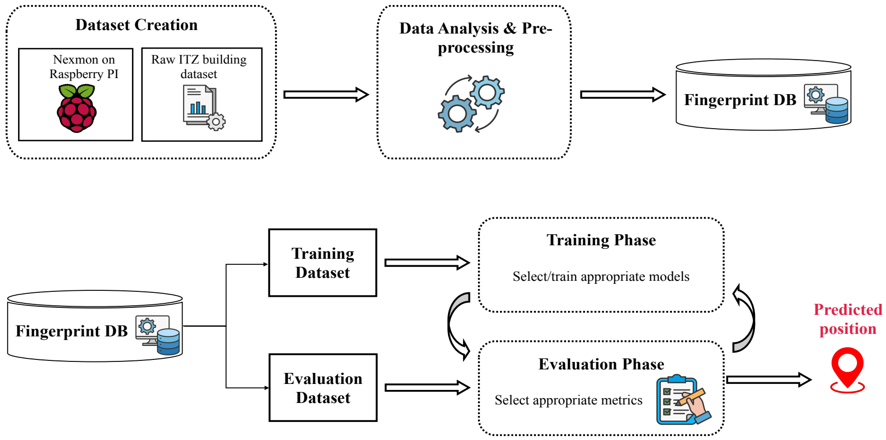

Fig. 1. Overall system architecture.

B. Data Collection

The collection of high-quality data is crucial for subsequent steps of preprocessing and the application of machine learning algorithms. This section provides details on the tool Nexmon CSI extractor, which is used to collect CSI and RSSI data in this work.

The Nexmon CSI Extractor [8], [9] is configured on a Raspberry Pi, allowing for the collection of CSI and RSSI data in a laboratory setting at the University of Passau. Introduced in 2019, it provides a good option to collect CSI data. It employs rooted Broadcom Wi-Fi chips found in multiple devices such as Nexus smartphones, Raspberry Pi boards, and Asus RT-AC86U routers. Custom firmware is employed to enable CSI data collection by listening on a specific User Datagram Protocol (UDP) socket. The Nexmon CSI Extractor offers support for 128 subcarrier groups with the highest resolution of 32 bits. Another significant advantage is its ability to simultaneously collect CSI and RSSI data.

The CSI and RSSI data was collected in a laboratory room located in the ITZ Building at the University of Passau. The room has a pentagon shape with an approximate area of 45 m^2. For each Reference Point (RP) (i.e., A RP is the point learned from the training phase, where the different RSS values were recorded.) inside the room, data was collected in four directions, north, east, south, and west. The RPs are *1 m* apart from each other and most of the outer ones are *0.45 m* apart from the walls. The numbering scheme of the RPs follows a vertical-horizontal structure, with the first (or first two, for three-digit numbers) digits representing the vertical position, and the last digit representing the horizontal position. The vertical axis ranges from 1 to 10 and the horizontal axis ranges from 1 to 5. This gives the room a matrix-like structure, which is useful for further data processing and position predictions.

For each RP, data was collected in four directions. The Raspberry Pi was positioned in front of a person's body, with a collection direction arrow pointing away from them. In each direction at each RP, 100 Wi-Fi Frames were collected. This resulted in a total of 400 Wi-Fi Frames per RP, and with 45 RPs amounting to a total of 18000 collected Wi-Fi Frames. The data is stored in .pcap files on a per direction, per RP basis, meaning that every .pcap file contains 100 Wi-Fi Frames. Data was collected over the course of one day (10h) by two researchers.

TABLE I
EXCERPT OF COLLECTED WI-FI FRAME STRUCTURE

Bytes	Type	Name	Description
1	uint8	RSSI	RSSI value in Two Complement Form
6	uint8[6]	Source Mac	Source Mac ID of the Wi-Fi Frame
variable	int16[]	CSI Data	Each CSI sample is 4 bytes with interleaved Int16 Real/Imaginary.

An excerpt of the structure of a collected Wi-Fi Frame can be found in Table I. For an IPS, the three shown variables in the table are of particular importance. The RSSI and CSI data fields describe the wireless characteristics of the Wi-Fi Frame with a specific Source MAC Address. While the RSSI and Source MAC field contain single values, the CSI data field contains both amplitude and phase information for each Orthogonal Frequency-Division Multiplexing (OFDM) subcarrier in complex form. In the setup of this study, data for 256 subcarriers (Bandwidth (80Mhz) * 3.2) is available. This raw network data is crucial for creating a high-quality fingerprint database. In the next step, this data must be cleaned and preprocessed, to prepare it for the machine learning algorithms.

C. Data Processing

To create the fingerprint database, data processing is a crucial step before training the ML models. In this paper, two fingerprint databases were created, one for the RSSI data and one for the CSI Amplitude data. The CSI Phase was

 ISBN: 978-1-68558-106-0

not considered since CSI Amplitude data was more stable and easier to process. For both RSSI and CSI Amplitude data, different preprocessing was performed. This section aims to give an insight into the used techniques and results of creating the datasets. To apply the preprocessing methods, the data was first extracted with the CSIKit library [10]. This library extracts the raw data of the .pcap files into a python environment, where further steps can be executed.

1) The CSI Amplitude dataset: it contains data for 45 RPs, and 1283 columns with information, shaping the dataset to be in a matrix form of 45x1283. With three columns containing position and direction information, the input dataset contains 57600 CSI Amplitude values. The mean CSI Amplitude values of all four directions were calculated for every RP, to make the dataset more robust. Also, the mean direction value of 2.5 was kept in the dataset to make this clearer.

The other columns each represent the CSI Amplitude values of a subcarrier for a specific Source MAC address. Subcarriers, that do not contain CSI Amplitude data like for example null subcarriers [11], are removed. The dataset is filtered using only the Source MAC addresses, which contain a reasonable amount of data. Not a Number (NaN) values are replaced with the minimum value of each column. To further improve the data quality, more preprocessing actions were done for the CSI Amplitude data. CSI data often contains noise and outliers that can distort the essential information. In this study, three filters are used for CSI Amplitude data processing: the running mean filter, a lowpass filter, and a Hampel filter [12]. For the running mean filter, a window size of 10 was chosen, the lowpass filter was configured to isolate frequencies below 10Hz and for the Hampel Filter, a window size of 10 and a significance of 3 was set. Figure 2 shows the effects of preprocessing the CSI Amplitude data for a specific RP.

2) The RSSI dataset: the size of this dataset is 45 RPs, with 8 columns containing data making it a matrix-like shape of 45x8. Three columns contain position and direction information, amounting to a total of 225 RSSI values. The other columns contain the actual RSSI values in dBm, the column names are the Source MAC addresses. Here, again only the Source MAC addresses with a reasonable amount of data were added, in order to reduce complexity of the dataset and therefore improve performance. As discussed earlier, for every direction of each Reference Point, there are 100 collected Wi-Fi Frames. For frames with the same Source MAC address, the mean RSSI values of them is used as value, further improving the robustness of the dataset. Frames with no Source MAC address information are ignored.

If no Wi-Fi Frame was collected for a specific MAC address at a RP, the resulting NaN values were replaced with the minimum possible value for RSSI, -100. With the now clean and preprocessed datasets, it is possible to train the machine learning models. The setup and results will be explored next.

III. ML Algorithms Application

In this section, the effectiveness of SVR, LSTM and CNN algorithms in leveraging both CSI amplitude and RSSI data, is evaluated and compared. With the preprocessed RSSI and CSI datasets of the previous section, the three different models are trained and evaluated. For better comparability, the horizontal and vertical positions are predicted separately, as the SVR approach allows for only one output. For the combined accuracy predicting both horizontal and vertical positions, the mean of the separate values is taken for the SVR approach. The LSTM and CNN implementations allow two outputs. To compare the results, the Mean Squared Error (MSE) is calculated for each model. For all three models, the CSI Amplitude and RSSI datasets are split into train and test datasets. Table II depicts the input dataset sizes for both datasets.

TABLE II
Sizes of input datasets

Value	Train Data (80%)	Test Data (20%)	Total
RSSI	180	45	225
CSI Amplitude	46080	11520	57600

A. Support Vector Regression (SVR)

SVR is a powerful machine learning technique used for solving regression problems [13]. SVR utilizes kernel functions to transform the input data into a higher-dimensional feature space, where linear regression is performed. For the environment of this study, the Radial Basis Function (RBF) kernel was chosen for transforming the input data into a higher-dimensional feature space. It can capture complex non-linear relationships between the input features and the output variable and therefore gave significantly better results than configurations with other kernels. Table III depicts the MSE results for the RSSI and CSI Amplitude datasets.

TABLE III
MSE predicting with SVR model

Value	Vertical		Horizontal		Vertical and Horizontal	
	Train	*Test*	*Train*	*Test*	*Train*	*Test*
RSSI	0.0110	0.0292	0.0531	0.0670	0.032	0.048
CSI Amplitude	0.0076	0.0419	0.0082	0.1156	0.007	0.078

For the RSSI dataset, the MSE was 0.032 for evaluating with the train data, and 0.048 when evaluating with unseen data. With the CSI Amplitude dataset, a the MSE for the train dataset was 0.007 and for the test dataset 0.078.

B. Long Short-Term Memory (LSTM)

In this study, a carefully designed and optimized LSTM model architecture was employed. The selected architecture comprises two layers. The first layer of the model is an LSTM layer. The LSTM layer consists of 50 units for the RSSI dataset, and 8 units for the CSI Amplitude dataset, allowing the model to capture and learn complex patterns in the input data. By leveraging its inherent memory cells, the LSTM layer can retain and utilize essential information from past observations to inform future predictions accurately. An overview of the configuration when predicting the vertical and the horizontal

 ISBN: 978-1-68558-106-0

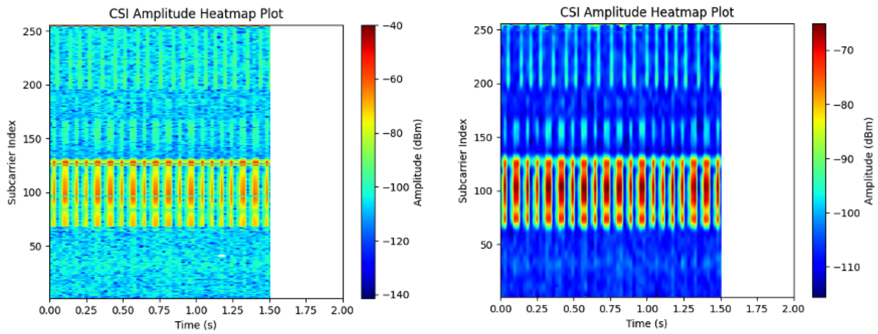

Fig. 2. Heat Maps of CSI Amplitude before (left) and after (right) preprocessing.

position for the LSTM approach is displayed in Table IV. The second layer of the model is a dense layer. The purpose of this layer is to consolidate the information extracted by the LSTM layer and make precise predictions. The number of units in the dense layer is determined based on the desired dimensions of the predicted outputs. For single-dimensional predictions, such as determining the location along a specific axis, a single unit is utilized. However, when predicting both the vertical and horizontal dimensions simultaneously, two units are employed to capture the multi-dimensional nature of the indoor positioning problem.

The MSE for the RSSI dataset was found to be 0.0415 when evaluating it with the training data, and 0.0691 when assessing it with unseen data. Regarding the CSI dataset, the MSE was 0.0008 for the training dataset and 0.0801 for the test dataset, as seen in Table V.

C. Convolutional Neural Networks (CNN)

In this work, a CNN-based ML model is employed for IPSs utilizing the CSI Amplitude and RSSI data. CNNs are particularly well-suited for tasks involving grid-like data, such as images or in this case, CSI Amplitude and RSSI data [14]. Two different architectures for RSSI and CSI Amplitude models were created, since the RSSI data differs a lot from the CSI Amplitude data. An overview of the architectures is depicted in Figure 3 for the RSSI dataset, and Figure 4 for the CSI Amplitude dataset.

MSE Results for both architectures are shown in Table VI. The RSSI based model achieves a MSE error of 0.031 for the training data, and 0.077 for unseen data. For the model trained with CSI Amplitude data, the MSE for training data is 0.0000007, and 0.066 for the test data. The MSE values were again calculated by getting the mean MSE of 30 trained models, to improve the robustness of the result.

Next, the results are transformed to represent distance error, compared, and discussed.

IV. Results and Discussions

To further compare the results of all three applied algorithms, the approximate average distance errors are computed, by rescaling the MSE to the length of the axes (i.e., 900 cm

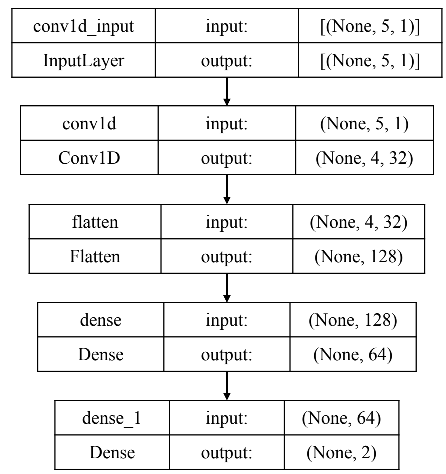

Fig. 3. CNN model architecture for RSSI data

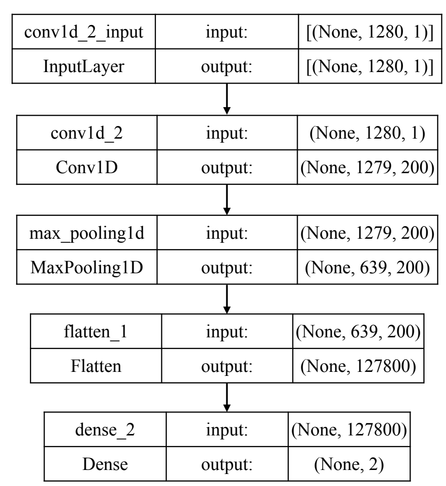

Fig. 4. CNN model architecture for CSI data

TABLE IV
LSTM MODEL CONFIGURATION FOR PREDICTING VERTICAL AND HORIZONTAL POSITION

Value	**LSTM-Layer units**	**Dense-Layer units**	**Optimizer**	**Epochs**	**Batch Size**
RSSI	50	2	Adam	1000	64
CSI Amplitude	8	2	Adam	200	32

TABLE V
MSE PREDICTING WITH LSTM MODEL

Value	**Vertical**		**Horizontal**		**Vertical and Horizontal**	
	Train	*Test*	*Train*	*Test*	*Train*	*Test*
RSSI	0.0185	0.0267	0.0496	0.1343	0.0415	0.0691
CSI Amplitude	0.0015	0.0363	0.0062	0.1059	0.0008	0.0801

for the vertical axis, 490 cm for the horizontal axis). This is done for each axis separately, as shown in Table VII and Table VIII, as well as when predicting the position with both axes, shown in Table IX.

A. Vertical Axis

Table VII focuses on predicting the vertical position using RSSI and CSI data. The distance errors are reported for both the training and testing sets.

The SVR algorithm achieves a distance error of 9.91 cm in the training set and 26.30 cm in the testing set when using RSSI data. When using CSI Amplitude data, the distance error reduces to 6.86 cm in the training set but increases to 37.79 cm in the testing set

The LSTM algorithm performs with a distance error of 16.66 cm in the training set and 24.05 cm in the testing set when using the RSSI data. However, when using CSI Amplitude data, the performance improves significantly, resulting in a distance error of only 1.42 cm in the training set and 32.70 cm in the testing set. The CNN algorithm demonstrates superior performance in predicting the vertical position. When using RSSI data, the distance error is 7.51 cm in the training set and 27.05 cm in the testing set. When utilizing CSI Amplitude data, the performance improves even further, achieving a remarkable distance error of only 0.0007 cm in the training set and 32.56 cm in the testing set.

Comparing the algorithms for vertical position prediction, the SVR has reliable results with the RSSI data, but struggles with the more complex CSI Amplitude data. The CNN approach gives comparable results as the SVR approach for the RSSI data, but outperforms both SVR and LSTM, as it consistently achieves lower distance errors with the CSI Amplitude data.

B. Horizontal Axis

Table VIII focuses on predicting the horizontal position using RSSI and CSI data. Like before, the distance errors are reported for the training and testing sets.

When using RSSI data, SVR achieves a distance error of 26.02 cm in the training set and 32.86 cm in the testing set. With CSI Amplitude data, the distance error decreases to 4.03 cm in the training set but increases to 56.65 cm in the testing set. LSTM performs with a distance error of 24.30 cm in the training set and 65.82 cm in the testing set when using RSSI data. When using CSI Amplitude data, the distance error improves slightly to 3.05 cm in the training set and to 51.93 cm in the testing set. CNN shows consistent performance in predicting the horizontal position. When using RSSI data, the distance error is 23.79 cm in the training set and 57.10 cm in the testing set. With CSI Amplitude data, the performance improves significantly, achieving a distance error of only 0.000009 cm in the training set and 43.24 cm in the testing set.

Comparing the algorithms for horizontal position prediction, CNN again outperforms SVR and LSTM in terms of distance errors, especially when utilizing CSI Amplitude data. For unseen RSSI data, SVR gives the best result.

C. Vertical and Horizontal Axis

Table IX presents the overall performance of the algorithms in predicting both vertical and horizontal positions using RSSI and CSI data.

The distance error for SVR when using RSSI data is 17.97 cm in the training set and 29.58 cm in the testing set. When utilizing CSI data, the performance improves with a distance error of 5.44 cm in the training set and 47.22 cm in the testing set. LSTM achieves a distance error of 20.48 cm in the training set and 44.93 cm in the testing set when using RSSI data. With CSI Amplitude data, the distance error improves to 2.23 cm in the training set and 42.31 cm in the testing set. CNN performs consistently well in predicting both vertical and horizontal positions. When using RSSI data, the distance error is 15.65 cm in the training set and 42.07 cm in the testing set. When utilizing CSI Amplitude data, the performance improves further, achieving a distance error of only 0.0003 cm in the training set and 37.90 cm in the testing set.

Comparing the algorithms for predicting both vertical and horizontal positions, CNN again demonstrates superior performance, achieving lower distance errors compared to SVR and LSTM, especially when utilizing CSI Amplitude data. Overall, based on the results of this study, the CNN algorithm consistently outperforms SVR and LSTM in terms of distance errors for predicting both vertical and horizontal positions in an indoor positioning system using CSI Amplitude data by about 20% and 10% respectively. Additionally, the performance of all algorithms, except SVR, generally improves when CSI data is used instead of RSSI data, highlighting the importance of considering CSI Amplitude data for accurate indoor positioning. For RSSI data only, SVR can give reliable

TABLE VI
MSE PREDICTING WITH CNN MODEL

Value	Vertical		Horizontal		Vertical and Horizontal	
	Train	*Test*	*Train*	*Test*	*Train*	*Test*
RSSI	0.0083	0.0300	0.0485	0,1165	0.031	0.077
CSI Amplitude	$7 * 10^{-7}$	0.0361	$2 * 10^{-8}$	0.0882	$7 * 10^{-7}$	0.066

TABLE VII
DISTANCE ERROR (CM) PREDICTING VERTICAL POSITION (MSE-BASED)

	RSSI		CSI Amplitude	
Algorithm	**Train**	**Test**	**Train**	**Test**
SVR	9.91	26.30	6.86	37.79
LSTM	16.66	**24.05**	1.42	32.70
CNN	**7.51**	27.05	**0.0007**	**32.56**

TABLE VIII
DISTANCE ERROR (CM) PREDICTING HORIZONTAL POSITION (MSE-BASED)

	RSSI		CSI Amplitude	
Algorithm	**Train**	**Test**	**Train**	**Test**
SVR	26.02	**32.86**	4.03	56.65
LSTM	24.30	65.82	3.05	51.93
CNN	**23.79**	57.10	**0.000009**	**43.24**

results as well, but the algorithm has limitations with the larger and more complex CSI Amplitude dataset.

V. CONCLUSIONS

In this study, an extensive dataset of CSI and RSSI data was meticulously collected within a controlled laboratory environment. The dataset serves as a solid foundation for future research endeavors in the field of IPSs. It encompasses crucial information, including the position with direction details, CSI Phase, CSI Amplitude, and RSSI measurements.

To assess the performance of the IPSs, three distinct ML algorithms were applied to the preprocessed datasets: SVR, LSTM and CNN. Notably, the integration of both CSI Amplitude and RSSI data yielded promising results, with all models achieving a mean distance error based on MSE of less than 50 cm, which is superior to all related works [14]–[16]. Among the individual metrics, SVR based solely on RSSI data demonstrated superior performance, attaining an MSE-based accuracy level of approximately 30 cm. Conversely, CNN, utilizing CSI Amplitude data, showcased the best results with an average MSE-based distance error of about 38 cm.

The findings of this work underscore the effectiveness of employing ML techniques, along with comprehensive preprocessing methodologies, to enhance the accuracy and reliability of IPSs. The results pave the way for future research to explore alternative algorithms, feature engineering techniques, and hybrid approaches to further improve the localization accuracy of IPSs in various indoor environments. By refining and expanding upon the methodologies established in this paper, IPSs can be further improved.

TABLE IX
DISTANCE ERROR (CM) PREDICTING VERTICAL AND HORIZONTAL POSITION (MSE-BASED)

	RSSI		CSI Amplitude	
Algorithm	**Train**	**Test**	**Train**	**Test**
SVR	17.97	**29.58**	5.44	47.22
LSTM	20.48	44.93	2.23	42.31
CNN	**15.65**	42.07	**0.0003**	**37.90**

REFERENCES

[1] R. Bajaj, S. L. Ranaweera, and D. P. Agrawal, "Gps: location-tracking technology," *Computer*, vol. 35, no. 4, pp. 92–94, 2002.

[2] W. F. Hassen and F. Najjar, "A positioning handoff decision algorithm for ubiquitous pedestrian navigation systems," pp. 487–494, 2016.

[3] Y. Gu, A. Lo, and I. Niemegeers, "A survey of indoor positioning systems for wireless personal networks," *IEEE Communications surveys & tutorials*, vol. 11, no. 1, pp. 13–32, 2009.

[4] C. Basri and A. El Khadimi, "Survey on indoor localization system and recent advances of wifi fingerprinting technique," in *2016 5th international conference on multimedia computing and systems (ICMCS)*. IEEE, 2016, pp. 253–259.

[5] W. F. Hassen, L. Brunie, A. Nechba, and H. Kosch, "Continuous indoor/outdoor pathway display algorithm for pedestrian navigation service," in *Proceedings of the 17th International Conference on Advances in Mobile Computing & Multimedia*, 2019, pp. 66–73.

[6] W. F. Hassen, F. Najjar, L. Brunie, H. Kosch, and Y. Slimani, "Smart pdr integration for ubiquitous pedestrian navigation service," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2017, pp. 1558–1563.

[7] P. Roy and C. Chowdhury, "A survey of machine learning techniques for indoor localization and navigation systems," *Journal of Intelligent & Robotic Systems*, vol. 101, no. 3, p. 63, 2021.

[8] M. Schulz, D. Wegemer, and M. Hollick, "Nexmon: The c-based firmware patching framework," *Res. Gate*, 2017.

[9] F. Gringoli, M. Schulz, J. Link, and M. Hollick, "Free your csi: A channel state information extraction platform for modern wi-fi chipsets," in *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, 2019, pp. 21–28.

[10] G. Forbes, S. Massie, and S. Craw, "Wifi-based human activity recognition using raspberry pi," in *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2020, pp. 722–730.

[11] G. L. Stuber, J. R. Barry, S. W. Mclaughlin, Y. Li, M. A. Ingram, and T. G. Pratt, "Broadband mimo-ofdm wireless communications," *Proceedings of the IEEE*, vol. 92, no. 2, pp. 271–294, 2004.

[12] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, "Data preprocessing for supervised leaning," *International journal of computer science*, vol. 1, no. 2, pp. 111–117, 2006.

[13] M. Awad and R. Khanna, *Efficient learning machines: theories, concepts, and applications for engineers and system designers*. Springer nature, 2015.

[14] W. F. Hassen and J. Mezghani, "Cnn based approach for indoor positioning services using rssi fingerprinting technique," in *2022 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 2022, pp. 778–783.

[15] Y. Zhang, C. Wu, and Y. Chen, "A low-overhead indoor positioning system using csi fingerprint based on transfer learning," *IEEE Sensors Journal*, vol. 21, no. 16, pp. 18 156–18 165, 2021.

[16] J. Wang and J. Park, "An enhanced indoor positioning algorithm based on fingerprint using fine-grained csi and rssi measurements of ieee 802.11 n wlan," *Sensors*, vol. 21, no. 8, p. 2769, 2021.