Article

HyTra: Hyperclass Transformer for WiFi Fingerprinting-based Indoor Localization

Muneeb Nasir [1], Kiara Esguerra [1], Ibrahima Faye [2], Tong Boon Tang [1], Mazlaini Yahya[3], Afidalina Tumian[3] and Eric Tatt Wei Ho[1]*

[1] Department of Electrical & Electronics Engineering, Universiti Teknologi PETRONAS, Perak, Malaysia.
[2] Department of Fundamental and Applied Science, Universiti Teknologi PETRONAS, Perak Malaysia.
[3] Petroliam Nasional Berhad, Malaysia;
* Correspondence: hotattwei@utp.edu.my;

Abstract: Indoor localization is an active area of research dominated by traditional machine-learning techniques. Deep learning-based systems have shown unprecedented improvements and have accomplished exceptional results over the past decade, especially the Transformer network within natural language processing (NLP) and computer vision domains. We propose the hyper-class Transformer (HyTra), an encoder-only Transformer with multiple classification heads (one per class) and learnable embeddings, to investigate the effectiveness of Transformer-based models for received signal strength (RSS) based WiFi fingerprinting. HyTra leverages learnable embeddings and the self-attention mechanism to determine the relative position of the wireless access points (WAPs) within the high-dimensional embedding space, improving the prediction of user location. From an NLP perspective, we consider a fixed order sequence of all observed WAPs as a sentence and the captured RSS value(s) for every given WAP at a given reference point from a given user as words. We test our proposed network on public and private datasets of different sizes, proving that the quality of the learned embeddings and overall accuracy improves with increments in samples.

Keywords: Deep Learning; Transformer; WiFi Fingerprinting; Indoor Localization; Classification; Received Signal Strength

1. Introduction

Location and proximity-based services have been utilized in many different environments and use-cases, from industrial warehouses to airports and hospitals; given the nature of their application (e.g., monitoring and navigation), precise position information is often required [1]. The localization market is expected to grow to $183.81 billion by 2027 [2] and approximately 70% of smartphone usage and 80% of data transmission occur in an indoor environment [3]. With the increasing demand for Internet of Things (IoT) capability in machines operating in these sites, the proliferation of smart devices both aids existing services and incentivizes the creation of new location-based services. Moreover, the recent Covid-19 pandemic warranted contact tracing in closed spaces. Localization with highly accurate position information was crucial to both track and minimize the spread of the virus.

Localization is the process of identifying the coordinates of a given device within a given operating environment, which can be either outdoor or indoor. The global navigation satellite system (GNSS) is the leading standard for outdoor localization. The main GNSS constellations and their operators are Global Positioning System (GPS) by the USA, GLONASS by Russia, Galileo by Europe and Beidou by China, with GPS being the more popular system. These individual satellite systems are well-established within outdoor localization and deliver accurate readings outdoors. However, in closed indoor environments, due to non-line of sight conditions, walls, ceilings, and furniture obstruct and further weaken (by absorbing) the already low-powered signal, making it difficult to accurately determine the device or user location. Thus, a satellite-based system is not a viable solution for indoor localization. These satellite systems are deemed ineffective due

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2023 doi:10.20944/preprints202303.0530.v1

to the substantial amount of signal attenuation and interference experienced by the signal as it propagates through a building, resulting in highly inaccurate indoor position estimates.

Indoor localization methods may be distinguished by the operating principle of the algorithm if it is a time, geometric or fingerprinting-based approach. Fingerprinting methods are the most popular approach, particularly WiFi fingerprinting based on received signal strength (RSS) [4]. This is no surprise due to the ubiquity of WiFi connectivity indoors. Ease and cost of implementation also play a major factor as existing WiFi infrastructure can be availed. Although WiFi fingerprinting using Channel State Information (CSI) is more precise than its RSS counterpart, an additional network card is required to obtain CSI data rendering it a less attractive solution.

We develop an RSS-based WiFi fingerprinting solution for indoor localization utilizing deep neural networks without employing addition CSI data. Our novel approach applies the Transformer neural network to transform WiFi RSS measured from WAP to precise indoor locations.The Transformer network [5] offers several advantages over other sequence processing neural networks like Recurrent Neural Networks (RNN) [6] and Long-Short Term Memory (LSTM) [7] and have been effectively deployed in applications for natural language processing [8–10], computer vision [11,12], and indoor localization using CSI [13,14]. Transformers process all pairwise embeddings of words in an input sequence rather than considering them sequentially and are more effective learning long-range dependencies in a sequence. In our work, we propose novel adaptations to and apply the Transformer network to RSS-based WiFi fingerprinting. The primary contributions of our paper are as follows:

- A novel solution for indoor localization utilizing the Transformer network for RSS-based WiFi fingerprinting. The proposed solution is purpose-built for a complex hierarchical environment (multi-building, floor, and room) and intended to deliver consistent and accurate results.
- Modifications to the transformer network to use learnable positional embeddings to improve the network's accuracy and provide insights into the WAP's position within the multi-building and multi-floor environments.

We provide evidence supporting the claim that deep-learning solutions benefit from larger datasets. We evaluate our solution on one public (UJIIndoorLoc [15]) and two private datasets, each differing in size.

2. Related Work

In this section, we give a general outline of the different methods employed for indoor localization and discuss existing RSS-based WiFi printing solutions utilizing deep learning methods. Indoor localization covers a broad spectrum of techniques, each requiring different data types to perform localization. We have categorized the different indoor localization methods based on the kind of data used:

- Time Approach: Time of Flight (ToF), Time Difference of Arrival (TDoA) and Return Time of Flight (RTOF) make use of the time it took for the signal to propagate to the receiver, the time intervals between each signal reception and the signal propagation round trip time, respectively. These methods, although accurate, are affected by clock synchronization, sampling rate, and signal bandwidth; ToF also requires line-of-sight for accurate performance [1,16].
- Geometric Approach: Angle of Arrival (AoA) and Phase of Arrival (PoA) rely on angle and phase estimation using antenna arrays to calculate the difference in arrival time and the distance between transmitter and receiver, respectively. These methods can deliver high accuracy but suffer degradation from non-line-of-sight, faded multipath signals and require complex hardware and algorithms to undo [1,16].
- Fingerprinting: captures an array of received signal strength (RSS) or channel state information (CSI) measurements at every reference point to build a collection of signals, which are used to compare with real-time measurements to pinpoint the user's location. This technique of comparing signals is effective as each location would have a unique fingerprint arising from the complex signal propagation within

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2023 doi:10.20944/preprints202303.0530.v1

the indoor environment. CSI offers better accuracy but is prone to noise and multipath fading; however, RSS is more easily obtained and cost-effective when compared to CSI, which requires an off-the-shelf network interface card (NIC) [1,14,16].

WiFi fingerprinting exploits machine learning and deep learning techniques as its main algorithm [17,18]. A database of WiFi RSS signals are collected at selected reference locations within the targeted indoor environment, and then a localization algorithm is trained on the collected data. During the inference stage, the trained algorithm is fed an array of newly collected RSS values from the user. The trained model predicts a location based on the similarity of the user-sent RSS array and the stored RSS data. This process is viewed as two separate phases: the offline and online phases, respectively. Figure 1 illustrates the two phases of WiFi fingerprinting.

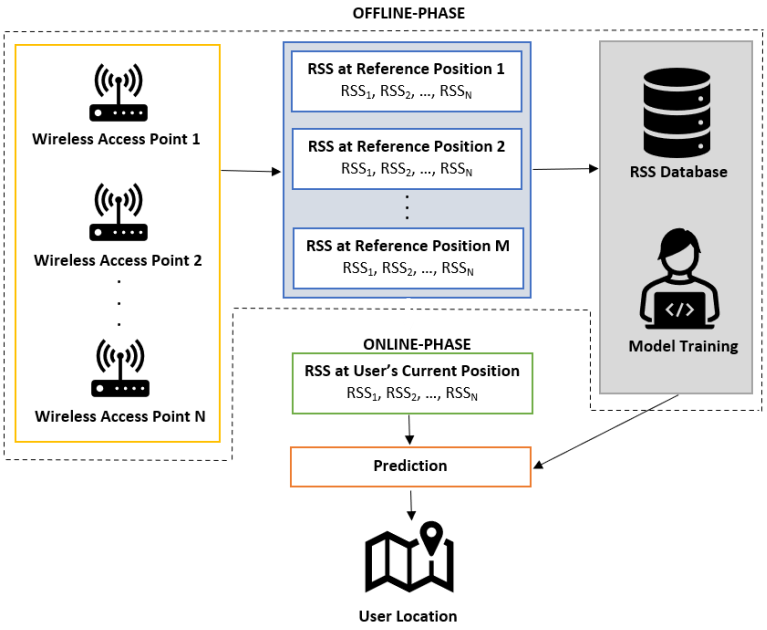

Figure 1. Standard RSS-based indoor WiFi fingerprinting method, with a trained machine learning or deep learning model as its prediction system.

Several classification and regression algorithms have been used for WiFi fingerprinting-based indoor localization, such as k-Nearest Neighbour (KNN) [15], Support Vector Machine (SVM) [19], Gradient Boosting [20], Multilayer Perceptron (MLP) [21], Convolutional Neural Network (CNN) [22,23], Recurrent Neural Network (RNN) [24,25], and Long Short-Term Memory (LSTM) [26] have been tested. These methods fall under machine learning and deep learning (a specialized subfield of machine learning), respectively. Machine learning sits at the intersection of computer science and statistics; it leverages statistical learning to extract patterns and make predictions from structured labelled data. It can be applied to unstructured data but requires some preprocessing [27]. In contrast, deep learning is a set of techniques enabling systems to learn complex behaviours by observing unstructured data, like text, images, and audio. Deep learning requires less human manipulation in the algorithm design stage, because the neural networks learn to extract essential features during training but require significantly more data than machine learning methods.

Stacked autoencoders (SAE) are commonly employed in deep learning-based RSS-based indoor localization. An autoencoder comprises an encoder and a decoder; the encoder learns a compressed representation of the RSS input, and the decoder reconstructs the original input from the compact representation. The stacked denoising autoencoder originally proposed by [28] allows dimension reduction and extraction of key features from the input, such as the sparse, noisy RSS input. Nowicki and

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2023 doi:10.20944/preprints202303.0530.v1

Wietrzykowski [29] pre-train the SAE to obtain an encoder effective at summarizing the RSS input; a deep neural network (DNN) is trained on top of the encoder to classify buildings and floors. This method achieves 92% for the building and floor prediction on the UJIIndoorLoc dataset (UJI). While [29] makes a combined prediction for building and floor, Kim et al. [30] propose to leverage the hierarchical nature of a multi-building, multi-floor complex to generate a label vector made of independent one-hot encodings of each identifier. Employing a similar network architecture and training strategy, Scalable DNNs achieves 99.5% and 91.26% accuracy on the UJI dataset using one-hot encoded hierarchical labels. Song et al. propose another SAE-based network, CNNLoc [23] but they attach a 1-dimensional convolutional neural network (CNN) to the pretrained SAE. Parameter sharing and sparse, local connections make CNNs less susceptible to overfitting. We believe that for these reasons, CNNLoc achieves an improved 100% and 96% accuracy for building and floor classification, respectively. Qin et al. employ a convolution-denoising autoencoder (CDAE) followed by another CNN for classification—dubbed Ccpos [31]. Original and noise-induced (Gaussian white noise) RSS data is fed to the CDAE to help reduce overfitting. Ccpos achieves accurate location estimates on the Alcala dataset [32] but not on the UJI dataset compared to existing methods, with an average positioning error of 1.05 meters and 12.40 meters on the respective datasets.

Laska and Blakenbach [33] introduced a custom label encoding scheme, output layer and loss function. Their proposed model, DeepLocBox, estimates a region in which the user is located via bounding boxes. DeepLocBox achieves a best score of 99.64% and 92.62% accuracy for building and floor classification on the UJI dataset out of 10 trials. In [34], Laska and Blakenbach propose multi-cell encoding learning to solve multi-task learning problems using a single forward pass network. Using a CNN backbone, the proposed network simultaneously classifies grid cells and does in-cell regression to achieve 95.3% accuracy in building and floor classification, also 7.18 meters mean positioning error. Recurrent neural networks (RNN) have been tested for RSS-based WiFi fingerprinting [35] and have shown to be a competitive solution to their DNN counterparts. The hierarchical RNN [35] uses the SAE to denoise and reduce the dimensionality of the RSS input. Due to the sequential nature of RNNs, hierarchical RNN effectively leverages the hierarchical underpinnings of multi-building multi-floor data to obtain 100% and 95.24% accuracy for building and floor classification on the UJI dataset.

3. Methodology

3.1 Data and Preprocessing

We evaluate our proposed Transformer model on the UJIndoorLoc public dataset (UJI) for comparison with existing techniques and a larger private dataset with approximately ten times more training and validation samples. We discuss our results from training and testing on both public and private dataset but only describe the public dataset in this section. The private datasets were constructed with similar details but differed in the number of samples, unique locations and WAP. UJIIndoorLoc [15] is the largest and most widely referenced dataset within the indoor localization literature and is easily accessible from the UC Irvine machine learning repository. The UJI dataset was compiled in 2013 at the Jaume I University, Castelló de la Plana, Valencian Community, Spain and is partitioned into a training and validation set comprising 19,937 and 1,111 records, respectively. Twenty (20) users on twenty-five (25) different android devices took measurements across three (3) buildings, each with four (4) floors on average, spanning a space of 110,000 m^2. The training and testing (addressed as validation in UJI) sets were generated four months apart to ensure data independence.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2023 doi:10.20944/preprints202303.0530.v1

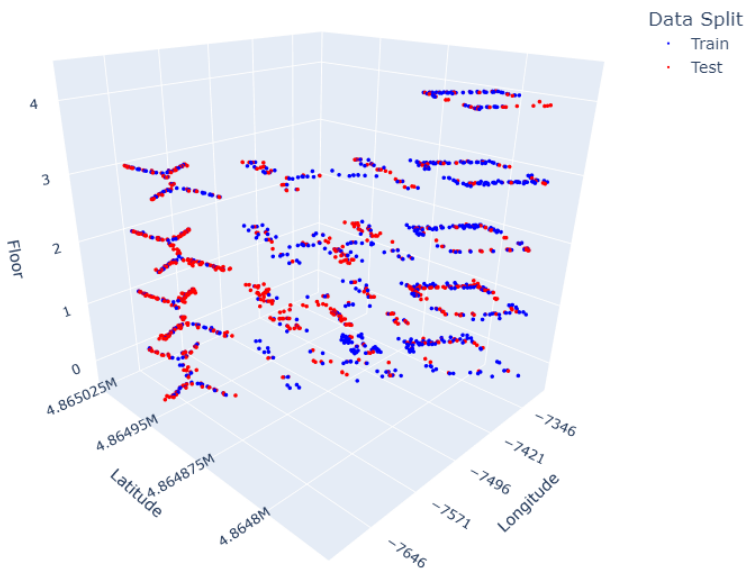

Figure 2. Training (blue) and testing (red) locations in UJIIndoorLoc; mapped by Longitude, Latitude and Floor.

The dataset contains RSS readings from 520 unique wireless access points ranging from -104 dBm (weakest detected signal) to 0 dBm (strongest detected signal). The 520 unique wireless access points are the summation of all the different access points encountered during the data collection process (training and testing). For any given measurement, the number of available access points is much less than the total number of access points–as shown in Figure 3. The authors [15] use positive 100 dBm to indicate the absence of an access point (out-of-range values).

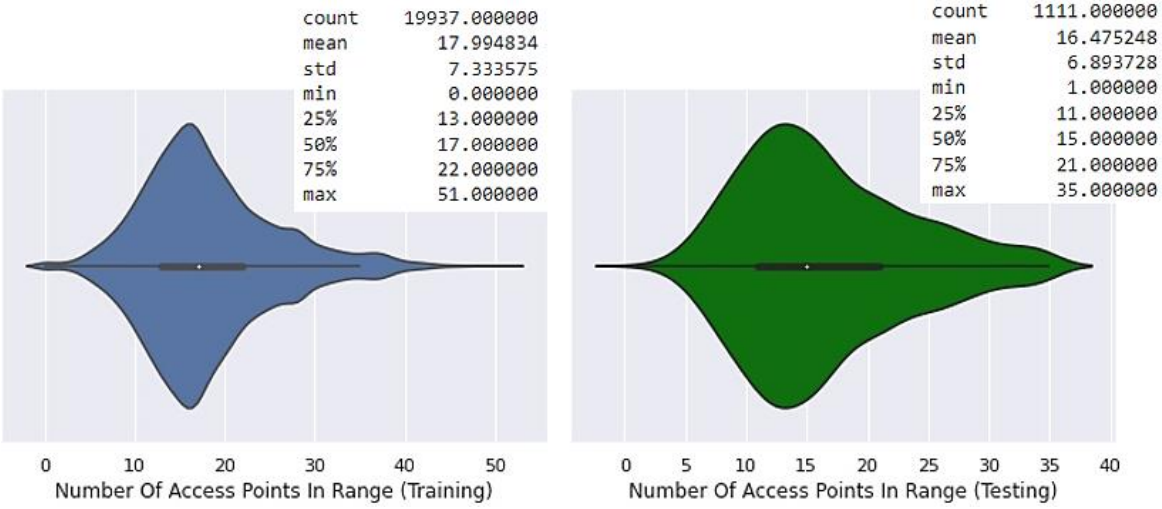

Figure 3. Distribution and statistics of the wireless access points used for both the training and testing data.

Table 1 summarizes the input and outputs used for fingerprinting. Five hundred twenty (520) wireless access points (WAPs) serve as input for each sample, and amongst the nine other identifiers, four are used as labels. Latitude and longitude values are labels for the regression task and are reported in meters. Building and floor labels are used for classification and have been assigned a numerical value corresponding to a specific building and floor. Building is set as either 0, 1 or 2, and the floors are originally designated as either 0, 1, 2, 3 or 4.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2023 doi:10.20944/preprints202303.0530.v1

Table 1. Input and label specification of fingerprinting data.

Input/Label	Fingerprinting Data	Description
Input	$\vec{R} = \{R_0, R_1, \dots, R_{520}\}$	RSSI values from 520 WAPs in dBm.
Classification Label	$B_k \ , k \in \{0, 1, 2\}$	Building ID, 3 unique buildings.
Classification Label	$F_u \ , u \in \{0, 1, \dots, 12\}$	Floor ID, 13 unique floors.
Regression Label	$long \in \{-7695.939, \dots, -7299.787\}$	Longitudinal value in meters.
Regression Label	$lat \in \{4864745.745, \dots, 4865017.365\}$	Latitudinal value in meters

The original dataset does not distinguish between floors across buildings i.e. the label for floor 1 in building 1, 2 and 3 are all the same (the label is 1). The lack of unique floor identity can be problematic for training a fingerprinting algorithm. We assign a unique label to every floor (seen in Tables 1 and 2). The five remaining identifiers are; Relative position, which indicates if the user was situated inside or at the entrance of a given space; UserID, PhoneID and Timestamp, equating to 529 attributes per sample. The remaining identifiers aided during data exploration but were not incorporated when training the HyTra network.

Table 2. Mapping of building and floor labels to unique floor labels.

Building	0				1				2				
Floor	0	1	2	3	0	1	2	3	0	1	2	3	4
Unique Floor	0	1	2	3	4	5	6	7	8	9	10	11	12

We transform the input and labels into a format that would allow the Transformer network to learn the best representation. Firstly, we replace the out-of-range RSSI values (+100 dBm) in the input with -110 dBm, as suggested by [29], since -104 dBm was the weakest signal observed within the training set. Then we apply zero-to-one normalization to obtain the scaled RSSI values:

$$Normalized\ RSS_{n,i} = \begin{cases} \dfrac{RSS_{n,i} - min}{-min}, & 0 \le i < 520 \\ & 0 \le n < 21{,}048 \\ & -110 \le RSSI_{n,i} \le 0 \end{cases} \quad (1)$$

where $RSS_{n,i}$ represents the n-th sample (training and test combined) and the RSS value from the i-th WAP, -110 dBm is the min value. Normalization is crucial as it maps features to the same scale, which helps with the trainability of the deep neural network. After normalizing the RSSI values, we may apply principal component analysis (PCA) to the input ($\vec{R}$, a one-dimensional array of length 520). As seen within the literature in the case of UJI, the noisy RSS input should be preprocessed to denoise and reduce dimensionality to mitigate overfitting. We test three input transformations (section *3.3.2*) to minimize overfitting for the UJI dataset, including PCA.

3.3 HyTra Model

We propose a multi-headed Transformer classification layer (one head per class) to account for the multiple classes (building, floor and room) within a complex indoor environment. We added building, floor and room specific learned position embeddings. Figures 4 and 5 illustrate our standard HyTra architecture for building-floor-room indoor localization and our UJI-specific HyTra implementation. We experiment with three different input transformations for our UJI-specific network to optimize our accuracy on the hold-out testing set.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2023 doi:10.20944/preprints202303.0530.v1

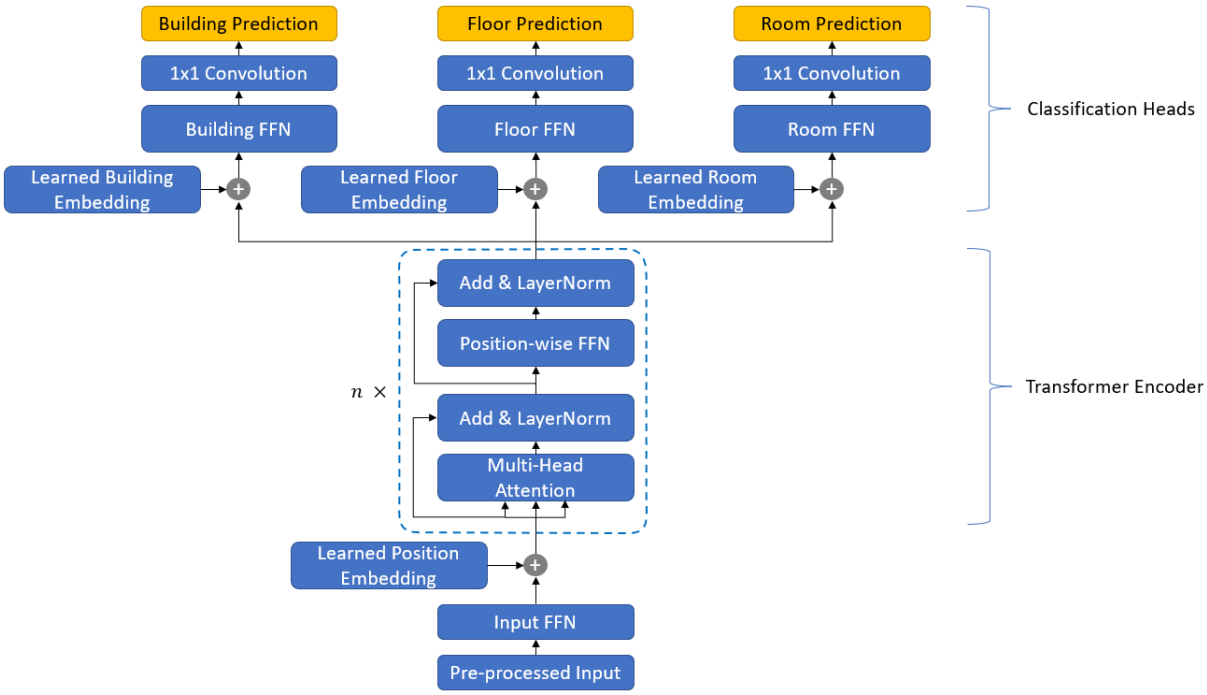

Figure 4. General HyTra network for the indoor localization classification task.

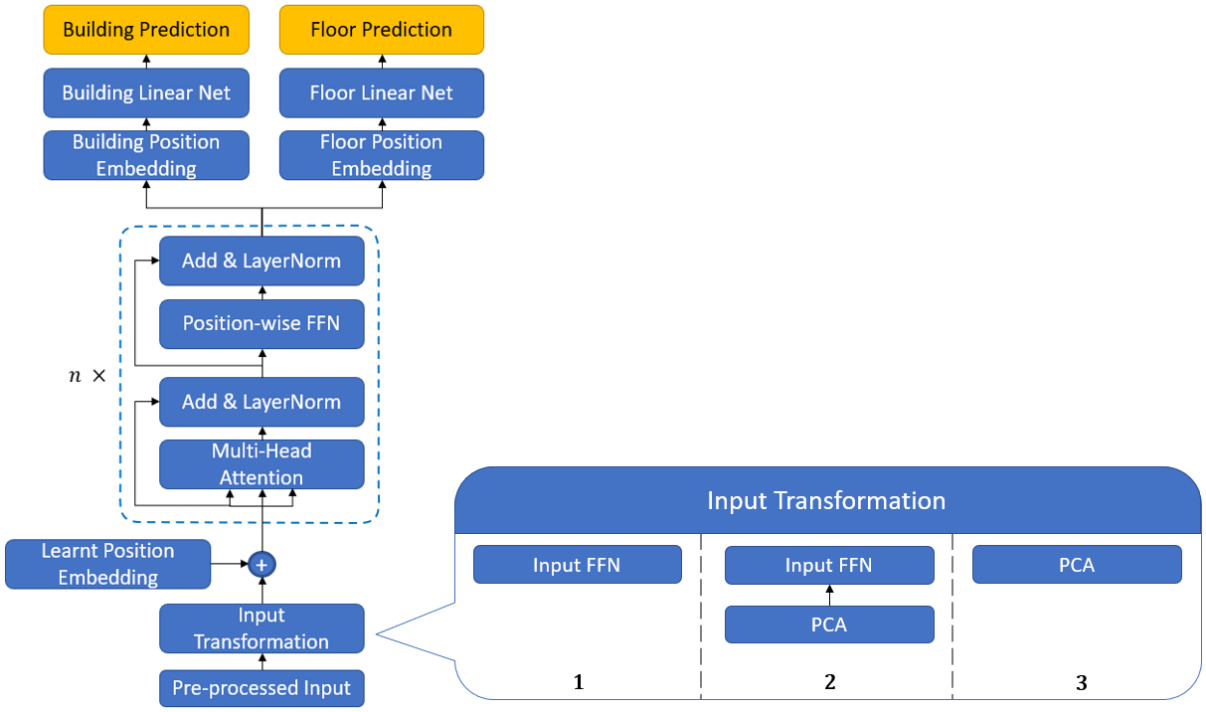

Figure 5. HyTra Network implementation for UJI dataset. Three different input transformations are tested to obtain the best testing set accuracy.

3.3.1 Model Input

After the RSSI data has undergone preprocessing, it is sequenced and batched; doing so results in an input shape: $Batch\ Size \times Sequence\ Length \times Features$. While our definition of $Batch\ Size$ (number of training samples per iteration) is fixed, $Sequence\ Length$ and $Features$ are not. We elaborate on the two ways we structure our input matrix in Table 3.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2023 doi:10.20944/preprints202303.0530.v1

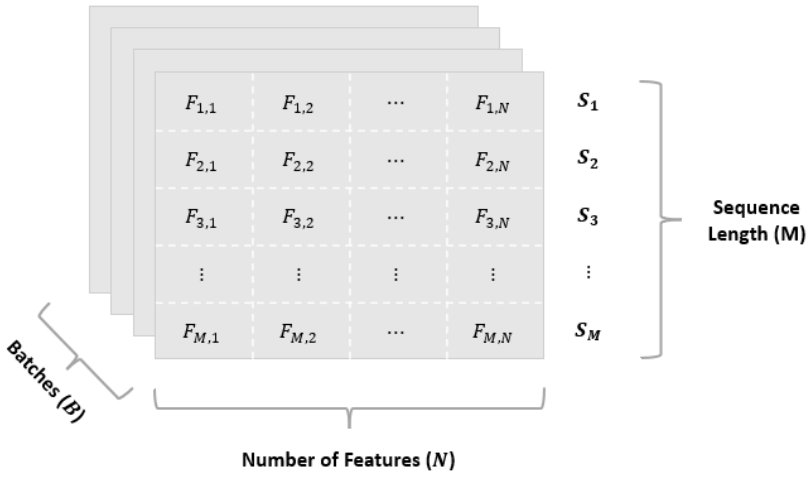

Figure 8. Standard batched input matrix representation. Shape: $B \times M \times N$.

Table 3. Dimension-wise descriptions for both input methods.

Input Method 1	**Sequence Length:** the total number of WAPs (520 for UJI) or principal components after applying PCA.
	Features: the number of RSS readings per WAP or scores per principal component.
	Objective: to allow the attention mechanism to weigh input based on the similarity of the features and allow the learnable position embedding to learn the relative position of each WAP. This method is preferred for larger datasets, as it would enable the network to compose a good representation of each WAP.
Input Method 2	**Sequence Length:** number of samples. A single sample can be used (sequence length of 1), but we recommend batching multiple samples chronologically to benefit from the attention mechanism.
	Features: RSS values from each WAP (520 for UJI) or scores of principal components after applying PCA.
	Objective: representing all of the WAPs or principal components as features are preferred for smaller datasets; this would circumvent the need to obtain an accurate input representation which is difficult on smaller datasets (like UJI by deep learning standards).

3.3.2 Input Transformation

The performance of a Transformer model is highly dependent on the quality of the input embeddings. A rich input embedding provides contextual and relational information or semantic meaning in the NLP case. Instead of individual words, we tokenize the WAP inputs and convert them to d-dimensional embedding vectors. We experiment with three input transformations and our two input methods to obtain the best representation of our input features, as seen in Figures 4 and 5. Firstly, we experiment with just a feed-forward network (FFN), composed of two linear layers with a ReLU activation in between and a dropout layer after. This transformation is primarily intended for input method 1, as there is only a single RSS value per WAP per sample; hence, we must scale up the feature dimension to increase the expressive ability of our network. This transformation is best suited to larger datasets as deep neural networks generally require lots of data to obtain high-fidelity representations.

Our second transformation method applies PCA to the dataset; then, it is structured, batched, and fed through a FFN. The intuition is that PCA reduces the data's dimensionality while retaining the most important components. As discussed in related works, existing techniques utilize methods or networks like the SAE to denoise the data before feeding it to their main classification algorithm.

Our last method only applies PCA to the data. Since we do not employ a FFN to upscale the feature dimension, we couple this transformation with our second input

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2023 doi:10.20944/preprints202303.0530.v1

method, where the PCA-transformed data (principal component scores) serves as the input features.

3.3.3 Position Embedding

Vaswani [5] originally generated position embeddings via a fixed sinusoidal function, while all our positional embeddings are learned through training. We employ learned positional embeddings as they are known to capture positional information and can model complex behaviours, which aid the training process [36]. Thus, we fix the order of WAPs in the input sequence and add learned positional embeddings before feeding the input to the Transformer and at each classification head (shown in Figure 4), which enables the embeddings to capture the order and relative positions of each WAP at each level of classification. Adding learnable embeddings to the transformed input at each classification head pushes vector representations of similar WAPs closer together in the high-dimensional embedding space. Note: applying PCA to reduce dimensions of the RSS input (shortened $sequence\ length$) maintains the underlying intuition of the learnable embedding; instead of expressing the relative position of WAPs, it represents the similarity of principal components.

3.3.4 Transformer Encoder

The Encoder contains two important sublayers, the multi-head attention layer and the pointwise feed-forward network (FFN). We employed the scaled dot product attention as the measure of similarity between input-pairs.

$$Attention(Q, K, V) = softmax\left(\frac{Q \cdot K}{\sqrt{d^K}}\right)V \quad (2)$$

where Q is the query matrix, K is the key matrix, V is the value matrix and d^K is the dimensionality of the key matrix. The query, key, and value matrices are derived from the input and go through a series of operations to compute the final attention-filtered value matrix (AFV) obtained using equation 2. The AFV matrix indicates to the network which WAPs to give the most importance. Then, the original position-embedded input is re-added to the attention-filtered value through a residual connection and normalized. Residual connections help mitigate the vanishing gradient problem by allowing the gradients to flow through the network without passing through non-linear activation functions. They also reinject positional information, which may fade as the input flows through complex layers. Multiple attention heads (single head refers to single-scaled dot-product operation) combine multiple attention mechanisms in parallel by concatenation and then are linearly transformed to match the original size.

The AFV is then fed to the pointwise feed-forward network, composed of two linear transformations with a ReLU activation in between and is applied to each position separately and identically. For the same reasons stated above, the attention-filtered value is added to the output of the FFN via residual connection and normalized before heading to the classification heads. The Encoder is stacked n times with identical layers, allowing the Encoder to model complex behaviour and extract hierarchical features.

3.3.5 Classification Heads

The Encoder obtains the most important features from the input, and then each classification head learns a mapping from these extracted features to their respective class. We add an embedding layer to each classification head to learn the relative position of each WAP at a building, floor and room level. Each classification head has a pointwise feed-forward network to aid the network in learning the complex mapping of AFV to building, floor and room. Lastly, a one-by-one convolution is applied over the input $sequence\ length$, extracting salient features across all WAPs and reducing the length to one. The resultant class of a given input is the index of the largest value in the output matrix, which is obtained using Argmax.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2023 doi:10.20944/preprints202303.0530.v1

3.4 Training Process

The training process is outlined in Algorithm 1.

Algorithm 1. Pseudo code of the proposed HyTra model

Input: $\vec{R}$, **Labels**: $B_k, F_l, long\ and\ lat$

Output: Predicted Location of User $(B_{predict}, F_{predict})$

if (B_k, F_l) are the given labels then
    Compute unique floor labels (F_u)
end for RSS in $\vec{R}$
    if $RSS == 100$ **then**
        Replace RSS with -110
end $\hat{R} = ZeroToOneNormalization(\vec{R})$
if Apply $PCA == True$ **then**
    $\hat{R} = PCA(\hat{R})$ **end**
$Train,\ Test,\ Validation = Split(\hat{R},\ stratify = F_u)$
if $Labels == (B_k, F_u)$ **then**
    Train HyTra classification model
    Finetune HyTra Classification model
    return $(B_{predict}, F_{predict})$
end

The hyperparameters for the HyTra network are listed in Table 4. We manually tune the values of the stated hyperparameters to settle on a configuration that delivers the best-performing solution. To accelerate the gradient descent algorithm, we use the AdamW [37] optimizer with a Cosine learning rate scheduler (with warmup), which combines both warmup and learning rate decay.

Table 4. Hyperparameters used to train HyTra.

Parameter	Value	Description
Learning Rate	0.0003	Step size of parameter update.
Scheduler	Cosine with Warmup	Adjusts learning rate during training.
Encoder Layers	2	Number of stacked identical encoder layers.
Attention Heads	8	Number of query, key and value pairs used for MHA.
Model Dimensions	256	Dimensionality of the transformed input features.
Dropout Rate	0.2	Fraction of inputs dropped during training.
Batch Size	128	Number of samples per iteration.
Optimizer	AdamW	Stochastic gradient descent method to update model parameters.
Loss Function	Cross-Entropy	Method to evaluate classification performance.
PCA Components	128	The number of components used to represent directions of maximum variance in the data.

3.5 System Overview

Figure 6 presents a system overview of the indoor localization algorithm using the proposed HyTra model for the classification task. Similar to other WiFi fingerprinting methods, our approach comprises of two phases: offline and online phase. Both phases require preprocessing the RSS data by replacing out-of-range values and applying zero-to-one normalization to bring RSS values to the same scale. During the offline phase, the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2023 doi:10.20944/preprints202303.0530.v1

HyTra model is trained using the preprocessed RSS data; the provided building labels and unique floor labels are used for classification, while zero-to-one longitude and latitude coordinates are used for regression. We trained and tested the HyTra network on Google Colab [38], a cloud-based Jupyter notebook environment with limited free GPU or unlimited paid access. Our setup utilized a Tesla T4 GPU, Intel (R) Xeon (R) CPU, Python-3.8.10 and Pytorch Lightning 1.9.4 for building, training and testing our HyTra network. The trained HyTra model predicts the user's location using RSS data collected and preprocessed during the online phase.

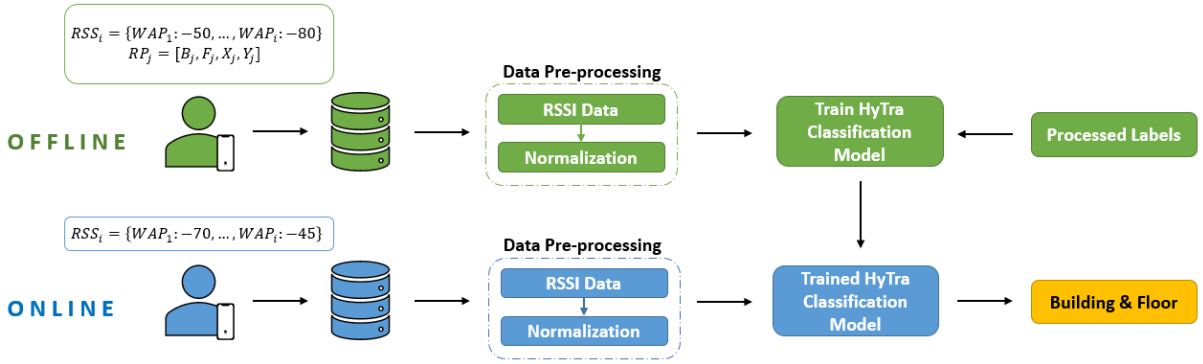

Figure 6: Overview of indoor localization scheme utilizing the proposed HyTra model for classification.

4. Results & Discussion

We evaluate and compare the classification performance of the proposed HyTra network on the public (UJI) and private datasets (SPOT and UTP) and compare our HyTra network against established deep learning-based techniques for indoor localization.

4.1 HyTra Comparison on Public and Private Datasets

Here we compare the classification performance of the HyTra network on the benchmark UJI dataset and our private datasets (SPOT and UTP). While the UJI dataset does not contain room-level labels within their testing set, our private datasets do. The difference in floor accuracy between the UJI and private datasets is significant.

Table 5. HyTra classification results using the UJI and two private (SPOT and UTP) datasets.

Datasets	Training Samples	Classification Accuracy		
		Building Accuracy	Floor Accuracy	Room Accuracy
SPOT (private)	165 K	100%	99.4%	97.9%
UJI (public)	17.9 K	100%	93.7%	-
UTP (private)	0.832 K	100%	97.1%	78.6%

[1] Classification accuracy reported based on the testing set.

Comparing classification accuracy for each data split on the UJI dataset (Table 6), we see the HyTra model overfit on the training set. There is a significant difference between training and testing floor accuracy in all three models. PCA with FFN (Input Transform 2) performs the worst out of the three input transformation methods, probably due to the FFN being unable to learn a good representation based on the low-rank estimate from PCA. PCA with the second input method obtains the best testing set accuracy on the UJI dataset. We believe this is due to the reduction of noise from applying PCA and not needing to learn a feature embedding vector (Input Method 2), treating the PCA-

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2023 doi:10.20944/preprints202303.0530.v1

transformed features directly as the input embedding. We notice a similarly large gap in performance on the testing set among others using UJI to test their deep learning-based methods [29].

Table 6. Classification results for each data split on the UJI dataset for HyTra with the different input methods and transformations.

Model	Dataset Split	Classification Accuracy	
		Building Accuracy	Floor Accuracy
HyTra (1,1)	Training	100%	100%
	Validation	99.85%	99.80%
	Testing	100%	94.33%
HyTra (1,2)	Training	99.22%	99.22%
	Validation	99.85%	99.10%
	Testing	99.91%	88.12%
HyTra (2,3)	Training	100%	99.24%
	Validation	99.35%	98.34%
	Testing	100%	96.47%

[1] HyTra (Input Method, Input Transformation), ordering as specified in the methodology section.

4.2 Comparison of Classification Results

Here we compare the classification accuracy of the proposed HyTra network with existing deep learning-based solutions discussed in related works. Compared with the best-performing deep learning models for indoor localization (Table 7), our standard HyTra using FFN Input Transform (2) and Input Method 1 is 1.7% less accurate than the best solution (CNNLoc). However, we obtain the highest floor accuracy when we directly treat the PCA-transformed input as feature embeddings.

Table 7. Comparison of classification results of existing methods and HyTra on the UJI dataset.

Deep Learning Methods	Classification Accuracy	
	Building Accuracy	Floor Accuracy
DNN [29]	-	91.10%
Scalable DNN [30]	99.5%	91.26%
Hierarchical RNN [35]	100%	95.23
2D-CNN (m-CEL) [34]	-	95.30%
CNNLoc [23]	100%	96.03%
HyTra (1,1)	100%	94.33%
HyTra (1,2)	99.91%	88.12%
HyTra (2,3)	**100%**	**96.47%**

[1] HyTra (Input Method, Input Transformation), ordering as specified in the methodology section.

5. Conclusion

We proposed a Transformer based Encoder-Only network using WiFi fingerprinting to classify complex indoor environments, dubbed HyTra. Our proposed solution leverages self-attention and learnable position embeddings to learn the relative position of WAPs at each classification level (building, floor and room). We generate unique floor labels to distinguish floors across the multi-building multi-floor complex. We propose two ways to structure input data and three different input transformations to experiment with to obtain the best input representation for large and small datasets. The private dataset (SPOT) results support our claim that model performance improves with increased training samples. Furthermore, our testing on the UJI dataset led us to test different input methods and transformations. The PCA-only transformation using input method two

obtained the best floor accuracy (96.47%) out of all the existing deep-learning-based techniques.

Author Contributions: "Conceptualization, writing—review and editing, E.T.W.H. , A.T. and M.Y.; methodology, M.N., A.T., M.Y., and E.T.W.H.; software, visualization, writing—original draft preparation, M.N.; investigation, validation, M.N. and K.E.; supervision, E.T.W.H., A.T. and I.F.; project administration, funding acquisition, A.T., E.T.W.H and T.B.T. All authors have read and agreed to the published version of the manuscript."

Funding: "This research and the APC was funded by Petroleum Research Fund (grant number E.025.FOF.02021.014) through Master Research Agreement between PETRONAS Research Sdn Bhd and Universiti Teknologi PETRONAS.

References

1. Mendoza-Silva, G.M.; Torres-Sospedra, J.; Huerta, J. A Meta-Review of Indoor Positioning Systems. *Sensors 2019, Vol. 19, Page 4507* **2019**, *19*, 4507, doi:10.3390/S19204507.
2. Location-Based Services (LBS) Market Size and Forecast - 2030 Available online: https://www.alliedmarketresearch.com/location-based-services-market (accessed on 9 March 2023).
3. Sithole, G.; Zlatanova, S. POSITION, LOCATION, PLACE AND AREA: AN INDOOR PERSPECTIVE. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **2016**, *III–4*, 89–96, doi:10.5194/ISPRS-ANNALS-III-4-89-2016.
4. Shang, S.; Wang, L. Overview of WiFi Fingerprinting-Based Indoor Positioning. *IET Communications* **2022**, *16*, 725–733, doi:10.1049/CMU2.12386.
5. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; N.Gomez, A.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *Adv Neural Inf Process Syst* **2017**, *2017-Decem*, 5999–6009.
6. Cho, K.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations Using RNN Encoder–Decoder for Statistical Machine Translation. In Proceedings of the Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2014; pp. 1724–1734.
7. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput* **1997**, *9*, 1735–1780, doi:10.1162/neco.1997.9.8.1735.
8. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models Are Few-Shot Learners. *Adv Neural Inf Process Syst* **2020**, *2020-December*, doi:10.48550/arxiv.2005.14165.
9. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. LLaMA: Open and Efficient Foundation Language Models. **2023**, doi:10.48550/arxiv.2302.13971.
10. Devlin, J.; Chang, M.-W.; Lee, K.; Google, K.T.; Language, A.I. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *Naacl-Hlt 2019* **2018**.
11. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. **2020**, doi:10.48550/arxiv.2010.11929.
12. Caron, M.; Touvron, H.; Misra, I.; Jegou, H.; Mairal, J.; Bojanowski, P.; Joulin, A. Emerging Properties in Self-Supervised Vision Transformers. *Proceedings of the IEEE International Conference on Computer Vision* **2021**, 9630–9640, doi:10.48550/arxiv.2104.14294.
13. Zhang, Z.; Du, H.; Choi, S.; Cho, S.H. TIPS: Transformer Based Indoor Positioning System Using Both CSI and DoA of WiFi Signal. *IEEE Access* **2022**, *10*, 111363–111376, doi:10.1109/ACCESS.2022.3215504.
14. Liu, W.; Cheng, Q.; Deng, Z.; Chen, H.; Fu, X.; Zheng, X.; Zheng, S.; Chen, C.; Wang, S. Survey on CSI-Based Indoor Positioning Systems and Recent Advances. *2019 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2019* **2019**, doi:10.1109/IPIN.2019.8911774.
15. Torres-Sospedra, J.; Montoliu, R.; Martinez-Uso, A.; Avariento, J.P.; Arnau, T.J.; Benedito-Bordonau, M.; Huerta, J. UJIIndoorLoc: A New Multi-Building and Multi-Floor Database for WLAN Fingerprint-Based Indoor Localization Problems. *IPIN 2014 - 2014 International Conference on Indoor Positioning and Indoor Navigation* **2014**, 261–270, doi:10.1109/IPIN.2014.7275492.
16. Zafari, F.; Gkelias, A.; Leung, K.K. A Survey of Indoor Localization Systems and Technologies. *IEEE Communications Surveys and Tutorials* **2019**, *21*, 2568–2599, doi:10.1109/COMST.2019.2911558.

17. Singh, N.; Choe, S.; Punmiya, R. Machine Learning Based Indoor Localization Using Wi-Fi RSSI Fingerprints: An Overview. *IEEE Access* **2021**, *9*, 127150–127174, doi:10.1109/ACCESS.2021.3111083.
18. Feng, X.; Nguyen, K.A.; Luo, Z. A Survey of Deep Learning Approaches for WiFi-Based Indoor Positioning. *Journal of Information and Telecommunication* **2022**, *6*, 163–216, doi:10.1080/24751839.2021.1975425.
19. Rezgui, Y.; Pei, L.; Chen, X.; Wen, F.; Han, C. An Efficient Normalized Rank Based SVM for Room Level Indoor WiFi Localization with Diverse Devices. *Mobile Information Systems* **2017**, *2017*, doi:10.1155/2017/6268797.
20. Singh, N.; Choe, S.; Punmiya, R.; Kaur, N. XGBLoc: XGBoost-Based Indoor Localization in Multi-Building Multi-Floor Environments. *Sensors* **2022**, *22*, doi:10.3390/s22176629.
21. Battiti, R.; Le, N.T.X.; Villani, A. Location-Aware Computing: A Neural Network Model for Determining Location in Wireless LANs. **2002**.
22. Liu, Z.; Dai, B.; Wan, X.; Li, X. Hybrid Wireless Fingerprint Indoor Localization Method Based on a Convolutional Neural Network. *Sensors 2019, Vol. 19, Page 4597* **2019**, *19*, 4597, doi:10.3390/S19204597.
23. Song, X.; Fan, X.; He, X.; Xiang, C.; Ye, Q.; Huang, X.; Fang, G.; Chen, L.L.; Qin, J.; Wang, Z. Cnnloc: Deep-Learning Based Indoor Localization with Wifi Fingerprinting. *Proceedings - 2019 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Internet of People and Smart City Innovation, SmartWorld/UIC/ATC/SCALCOM/IOP/SCI 2019* **2019**, 589–595, doi:10.1109/SMARTWORLD-UIC-ATC-SCALCOM-IOP-SCI.2019.00139.
24. Lukito, Y.; Chrismanto, A.R. Recurrent Neural Networks Model for WiFi-Based Indoor Positioning System. *2017 International Conference on Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS)* **2017**, 121–125, doi:10.1109/ICON-SONICS.2017.8267833.
25. Hoang, M.T.; Yuen, B.; Dong, X.; Lu, T.; Westendorp, R.; Reddy, K. Recurrent Neural Networks for Accurate RSSI Indoor Localization. *IEEE Internet Things J* **2019**, *6*, 10639–10651, doi:10.1109/JIOT.2019.2940368.
26. Chen, Z.; Zou, H.; Yang, J.F.; Jiang, H.; Xie, L. WiFi Fingerprinting Indoor Localization Using Local Feature-Based Deep LSTM. *IEEE Syst J* **2020**, *14*, 3001–3010, doi:10.1109/JSYST.2019.2918678.
27. What Is Deep Learning? | IBM Available online: https://www.ibm.com/topics/deep-learning (accessed on 22 February 2023).
28. Ca, P.V.; Edu, L.T.; Lajoie, I.; Ca, Y.B.; Ca, P.-A.M. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion Pascal Vincent Hugo Larochelle Yoshua Bengio Pierre-Antoine Manzagol. *Journal of Machine Learning Research* **2010**, *11*, 3371–3408.
29. Nowicki, M.; Wietrzykowski, J. Low-Effort Place Recognition with WiFi Fingerprints Using Deep Learning. **2017**, doi:10.1007/978-3-319-54042-9_57.
30. Kim, K.S.; Lee, S.; Huang, K. A Scalable Deep Neural Network Architecture for Multi-Building and Multi-Floor Indoor Localization Based on Wi-Fi Fingerprinting. *Big Data Anal* **2018**, *3*, doi:10.1186/s41044-018-0031-2.
31. Qin, F.; Zuo, T.; Wang, X. Ccpos: Wifi Fingerprint Indoor Positioning System Based on Cdae-Cnn. *Sensors (Switzerland)* **2021**, *21*, 1–17, doi:10.3390/s21041114.
32. Montoliu, R.; Sansano, E.; Torres-Sospedra, J.; Belmonte, O. IndoorLoc Platform: A Public Repository for Comparing and Evaluating Indoor Positioning Systems. *2017 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2017* **2017**, *2017-January*, 1–8, doi:10.1109/IPIN.2017.8115940.
33. Laska, M.; Blankenbach, J. Deeplocbox: Reliable Fingerprinting-Based Indoor Area Localization. *Sensors* **2021**, *21*, 1–23, doi:10.3390/s21062000.
34. Laska, M.; Blankenbach, J. Multi-Task Neural Network for Position Estimation in Large-Scale Indoor Environments. *IEEE Access* **2022**, *10*, 26024–26032, doi:10.1109/ACCESS.2022.3156579.
35. Elesawi, A.E.A.; Kim, K.S. Hierarchical Multi-Building And Multi-Floor Indoor Localization Based On Recurrent Neural Networks. *CoRR* **2021**, *abs/2112.12478*.
36. Wang, Y.-A.; Chen, Y.-N. What Do Position Embeddings Learn? An Empirical Study of Pre-Trained Language Model Positional Encoding. **2020**.
37. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. *7th International Conference on Learning Representations, ICLR 2019* **2017**, doi:10.48550/arxiv.1711.05101.
38. Google Colab Available online: https://research.google.com/colaboratory/faq.html (accessed on 9 March 2023).