INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 5, Issue 3)
Available online at: www.ijariit.com

Gesture recognition using RF signals

Chandrakanth R.
chandrakanth.rjsc@gmail.com
Dr. Ambedkar Institute of Technology, Bengaluru, Karnataka

Bharath K.
bharathk59@gmail.com
Dr. Ambedkar Institute of Technology, Bengaluru, Karnataka

Bhargava U. G.
bhargavaug97@gmail.com
Dr. Ambedkar Institute of Technology, Bengaluru, Karnataka

Chinmaya G.
gchinmaya3398@gmail.com
Dr. Ambedkar Institute of Technology, Bengaluru, Karnataka

Girija S.
girija.s@dr-ait.org
Dr. Ambedkar Institute of Technology, Bengaluru, Karnataka

ABSTRACT

Gesture recognition is gaining increasing importance in Human-Computer Interaction (HCI). Gesture-based interaction serves as a convenient and natural means for users to interact with computers. However, accurate detection and recognition of human actions is still a big trial that attracts lots of research efforts due to the difficulties related to the human body parts and the difficulty in sensing their actions correctly such as human clothes and their negative consequence on the detection accuracy and the surrounding environmental conditions. Vision centred human activity analysis by means of computer vision as the original answer is still having its limits that are connected to the inability to detect whatsoever happening behind the walls or in the dim places and the uncomfortable feeling of people with cameras all over the place. This paper proposes a methodology that will use RF signals to overcome all the disadvantages of the mentioned methods.

Keywords— *Gesture recognition, RF signals, Wi-Fi, RSSI, LSTM RNN*

1. INTRODUCTION

Generally, human identification is based on one or more intrinsic physiological [1] [2] [3] [4] or behavioral [5] [6] [7] differences, which either is associated to shape of the body e.g., fingerprint, face characters, iris, palm print or specific performance patterns of a person e.g., gait, voice rhythms, typing. It plays an important role in the region of pervasive computing and also human-computer interaction. Currently, fingerprint- [2], iris- [4], and vein-based approaches [8] have been effectively deployed in automatic human identification systems. However, these systems necessitate the user to be near to the sensing device for accurate identification. Researchers similarly made numerous attempts to cultivate approaches for behavioural biometrics (mainly gait analysis) using cameras, wearable sensors, or radars [7]. However, the vision-based methods only work on line-of-sight exposure and rich-lighting environments, which as well cause privacy concerns. The low cost of 60 GHz radar methods can only offer an operation range of tens of centimetres, and the devices are not broadly installed in our daily life. Finally, wearable sensor-based methods require people to wear extra sensors. The gesture made with fingers is chiefly crucial to interact with mobile and wearable devices and to perform finger control in smart home and mobile gaming. Google's Soli radar chip [9], for example, is newly developed for the wearable to recognize finger gestures. Existing gesture recognition solutions primarily depend on dedicated sensors worn by the subject or cameras installed in the environment. These systems either require significant deployment effort or incur non-negligible cost.

Considering the issues mentioned above, we propose a new machine learning approach centred on LSTM RNN for recognizing gestures in unmodified smartphones, based on artificially induced data traffic between a smartphone and a Wi-Fi Access Point (AP). The key contributions of this paper are:

- We exhibit that Wi-Fi RSS can be used to identify hand gestures near smartphones (see figure 1) using a blend of machine learning techniques and conventional signal processing algorithms.
- We do several experiments under several circumstances to validate the gesture sorting performance of the proposed approach, and it against the state of-the-art machine learning methods.

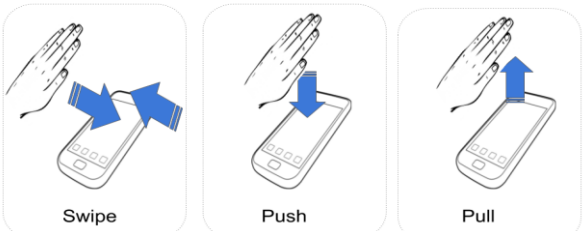

Fig. 1: The hand gestures considered in this paper

2. LITERATURE REVIEW

2.1 Human Identification

Researchers have made numerous efforts to develop methods for human identification, which can be grouped into two categories: physiological feature-based approaches and behavioural feature-based approaches.

Physiological features: Fingerprint [2], iris [4], and vein authentications [8] have been fruitfully employed in automatic human identification schemes. Furthermore, Duta [1] utilized the hand-shape to distinguish humans. Chellappa et al. [10] proposed a technique to recognize human-centred on-face recognition. However, these schemes require the subject to be nearby to the sensor for accurate identification.

Behavioural features: There are also some studies about behavioural biometrics, especially in gait analysis. Since human walking gesticulation results in dynamic rhythm and self-sustaining due to the integrated signals produced from the spinal cord and sensory feedback, it has exclusive characteristics for each individual. Little and Boyd [6] created a model-free description of instantaneous motion and utilized it to distinguish individuals by their gait. The work is centred on camera sensing, which needs a line of sight and enough lighting, and it also causes privacy issues. Nickel et al. [7] used HMM to recognize human gait centred on accelerometer data, and it needs users to wear relevant sensors.

2.2 Gesture Recognition

Existing gesture-recognition systems can be classified as vision-based, infrared-based, electric-field sensing, ultrasonic, and wearables. The Xbox Kinect, Leap Motion, Point- Grab, and CrunchFish utilize advances in cameras and computer vision to assist gesture recognition. Visible-light base approaches, however, by definition, cannot work in non-line-of-sight circumstances. The Samsung Galaxy S4 introduced an "air gesture" feature that utilizes infrared detectors and emitters on behalf of gestures, but is known to be sensitive to lighting conditions [11] and is limited to line-of-sight. Ultrasonic schemes such as SoundWave [12] transmit ultrasound waves and analyze them for gesture recognition. However, we are not aware of ultrasonic gesture systems that operate in non-line-of-sight scenarios. Finally, preceding work on inertial sensing and additional on-body systems need instrumenting the human body with detecting devices [13] [14]. In contrast, we concentrate on gesture recognition minus such instrumentation. Prior works in gesture recognition primarily rely on pre-installed infrared and depth cameras (e.g., leap motion, Kinect,) [15] [16] [17] or dedicated sensors (e.g., gloves, motion sensors, RFID) which are worn by user [18] [19]. These methods, however, require significant deployment overhead and sustain non-negligible cost. In addition, the camera-based method cannot work in non-line-of-sight (NLOS) scenarios. Some recent works rely on motion sensors available in today's smartphones to perform gesture recognition, or allow the user to write in the air (e.g., [17]). Others make use of such sensors on wearables like smartwatches [18], armbands [19], wristbands [15], and rings [20].

Radio frequency signals, notably Wi-Fi signals, have been lately used for sensing and distinguishing human activities [12], [13]. For instance, customized hardware-centred dynamic sensing solution is introduced in [14] utilizing transmit and receive arrays/antennas with Doppler /Fourier analysis of RSS data. They attained the identification accuracy of 94% to categorize nine full-body gestures in a home environment. Similarly, custom hardware-based methods (e.g. [15]) remain limited to the application environments and the resources. Nevertheless, antenna-array based methods find sophisticated new methods like seeing through walls using radio frequency signals [16].

In [11], the authors utilized both Wi-Fi RSS and Channel State Information (CSI) to distinguish hand gestures utilizing signal conditioning and thresholding-based gesture identification algorithm and achieved a distinguishing accuracy of 91% on a personal computer. Note, the CSI delivers detailed channel features as well as the sub-carrier level phase data, but it is sustained only by a very small set of Wi-Fi devices. Several different works such as [17]–[20] also suggested CSI centred solutions for gesture or activity recognition; however, they are exposed to similar limitations.

K-Nearest Neighbors (K-NN) centred classifiers have been widely utilized to recognize gestures from the RSS data. In [5], the author's utilized statistical features which are window-based (e.g. mean, maximum, number of peaks, variance, etc.) applied to a K-NN classifier for identifying hand gestures on a smartphone. They attained accuracies of 90% (with K=5, four hand gestures) and 50% (with K=20, 11 hand gestures) respectively. However, their solutions require modified device firmware, root access to OS, and devoted applications that limit smartphone's Wi-Fi traffic.

A DWT (Discrete Wavelet Transform) centred method transforms the RSS data intro three primitive signals: falling edges, rising edges and pauses. It attained high accuracy (90%) by utilizing a classifier that associates the sequences of primitive signals for a set of pre-defined rules. However, such a method needs extensive computation capabilities and high-frequency sampling of RSS.

The literature works can be concluded in four ways:

- we do not kind any modifications to the prevailing hardware or software applications of the phone;
- we introduce a novel traffic induction method to enable high-frequency RSSI measurements;
- we use custom albeit modest signal processing techniques and an efficient LSTM-RNN machine learning method to categorize over-the-air hand gestures.

To the best of our knowledge, very little mechanisms use deep learning or neural network-based methods to radio signal-based activity/gesture classification. A CNN (Convolutional Neural Network) for categorizing user driving behaviours centred on the narrow-band radio frequency signals and achieved 88% accuracy was used. On a higher level, the CNN is best for image processing owing to its nature of identifying patterns across space (inside the data), whereas the RNN is best for time-series /speech processing because of its between-data (across sequence) recognition abilities. Thus, we select RNN as the core machine learning algorithm in our solution.

3. THEORETICAL BACKGROUND

3.1 Radio Frequency (RF) Wave Propagation

An RF signal propagating through a medium is subject to various environmental factors that influence its characteristics. In the nonexistence of nearby obstacles, the signal strength will be decreased by the free-space path loss (FSPL) caused through the spreading out of the signal energy in space. Friis transmission equation (Equation 2.1) describes the relation between power sent by a transmitting antenna and the one received by a receiving antenna [21]. It also explains the impact of the FSPL in the signal power. The received power exists, such that it is

inversely proportional to, the square of the distance between transmitter and receiver (R) and also inversely proportional to the square of the signal frequency ($\frac{C}{\lambda}$), where C is spthe eed of light and λ is signal wave length.

$$P_r = P_t G_r G_t \left(\frac{\lambda}{4\pi R}\right)^2 \quad (1)$$

In the above equation, P_t and G_t are the transmitter output power and its antenna again, P_r and G_r are the receiver input power and its antenna gain, R is the distance between antennas and λ is the signal wavelength.

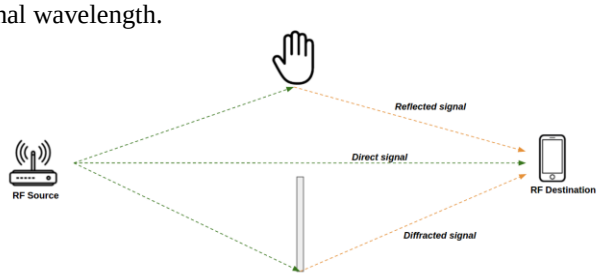

Fig. 2: Reflection and diffraction illustration

The RF signal can be also absorbed by the medium which it propagates in and causes a reduction in its signal strength. This reduction is proportional to RF signal frequency and the conductivity of the propagation medium [22]. Consequently, metal objects and the human body absorb RF signal power more than wooden objects.

The path of the signal can also be obstructed by surrounding objects by reflecting, refracting or diffracting the original signal [23]. Figure 2 and 3 illustrates the impact of the different factors affecting the RF signal propagation path.

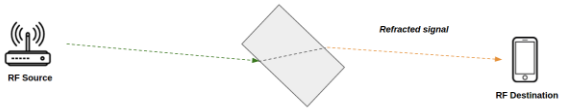

Fig. 3: Refraction illustration

3.2 Wi-Fi RSSI based gesture recognition

Rajalakshmi et al. in [24] utilized the Wi-Fi RSSI and Channel State Information (CSI) to recognize hand gestures. Compared to RSSI, CSI, which is defined by the IEEE 802.11n standard, provides detailed radio channel information consisting of both the signal strength and phase information for each sub-carrier in the radio channel [25]. The authors used consecutive windows of 300 ms to extract features consisting of the signal peaks, the peaks count and the slopes of the peaks after subtracting the window-average signal strength from all the samples within the window. The detection algorithm consisted of comparing each window extracted features to a set of four predefined feature values, each corresponding to a one hand gesture. The system attains classification correctness of 91% when tested on a notebook PC. Their solution has two key limitations: first it requires a Wi-Fi transmitter injecting Wi-Fi packets at a constant rate, and second, it relies on the CSI information that is supported by very few Wi-Fi devices (as per our knowledge, only Intel's Wi-Fi Link 5300 network adapter [26]).

In, Sigg et al. extended their work in [27] by using more features and verifying the system on hand gestures identification task. Specifically, like features, they used the average and the variance, the number of peaks within 10% of the maximum and the fraction between the mean of the 1*st* and 2*nd* half of the window. An accuracy of 50% was achieved when a K-NN (K=20) network was trained to classify between 11 hand gestures (random guess is 9.09%).

A passive hand gesture recognition system using Wi-Fi RSSI on unmodified devices was recently introduced in. The recognition algorithm involved the first denoising the raw RSSI measurements by means of the Discrete Wavelet Transform (DWT). Using the denoised version of the signal, further wavelet analysis was performed to translate the signal into a stream of primitives formed out of three types of signal primitives: growing edges, falling edges and pauses. The classification was then done by comparing the primitive streams with pre-defined streams corresponding to the hand gestures. When tested on a notebook PC, the system achieved an identification accuracy of 96% classifying between seven hand gestures. Such a system requires high-frequency sampling of RSSI and also needs a lot of computation power.

4. PROBLEM FORMULATION AND MODELLING

4.1 Formulation

The problem of identifying hand gestures from Wi-Fi RSSI values can be viewed as a classification problem, where the goal is to learn a mapping from the input Wi-Fi RSSI sequence x to a hand gesture y.

$$x \rightarrow y \quad (2)$$

Where $x = [x^{(1)}x^{(2)} \ldots x(t) \ldots x^{(\tau)}]^T$, $x^{(t)} \in R$, $y \in \{0,1,\ldots,K\}$, τ is the RSSI sequence length and K is the number of gestures recognizable by the mapping.

In a classification setting, rather than estimating a single value for the output variable y, instead it is most common to estimate the probability distribution over the output variable y conditioned on input x; precisely $P(y|x)$. This conditional probability can be estimated using a distribution family parameterized by a variable θ. This mapping can be expressed as:

$$x \rightarrow P(y|x;\theta) \quad (3)$$

Assume a dataset of m sample gestures that is formed from the inputs $X = [x_1\, x_2 \ldots x_i \ldots x_m]$ and their corresponding outputs $Y = [y_1\, y_2 \ldots y_i \ldots y_m]$. A maximum likelihood (ML) method can then be used to find a good estimation of θas.

$$\theta_{ML} = arg_\theta max\, P(Y|X;\theta) \quad (4)$$

And assuming the dataset sample gestures are independent and collected following the same procedure, we could assume that the dataset is independent and identically distributed (i.i.d.). Equation 4 can be revised as follows:

$$\theta_{ML} = arg_\theta max \prod_{i=1}^{m} P(yi|xi;\theta) \quad (5)$$

The above probability product can become too small and hence make the problem computationally unstable. This can be resolved by taking the logarithm of the likelihood, which transforms the product of probabilities into summation (the logarithm does not change the arg max operation):

$$\theta_{ML} = arg_\theta max \sum_{i=1}^{m} log\, P(yi|xi;\theta) \quad (6)$$

This estimate of θ can be defined as minimizing a loss function L (also referred to as cost) defined as below

$$L = \sum_{i=1}^{m} log\, P(yi|xi;\theta) \quad (7)$$

This loss is called *negative log-likelihood (NLL)*.

A form of a recurrent neural network (RNN) is considered to model the conditional probability $P(y_i|x_i;\theta)$. Using the negative log-likelihood loss, a maximum likelihood estimation of the RNN parameters θ can be found using a gradient-based

optimization procedure. In this thesis, the RNN model is trained using a variant of the Stochastic Gradient Descent (SGD) algorithm [30].

4.2 Artificial Neural Networks

4.2.1 Feed Forward Neural Networks (FFNN): Recurrent neural networks are a special form of Artificial Neural networks (ANN). ANNs are powerful general function approximators inspired by the working of the brain neurons.

The most basic form of ANN is *fed forward neural networks* (FFNN), which can be viewed as a layered directed acyclic computation graph, as depicted by Figure 4. A typical FFNN will be formed of an *input layer*, zero or more than zero *hidden layer(s)* and an *output layer*. The network input *x* will get processed by one layer at a time, starting at the input layer. The output of each layer forms the input of the following layer. The output of the last layer (output layer), correspond to the network output *y*.

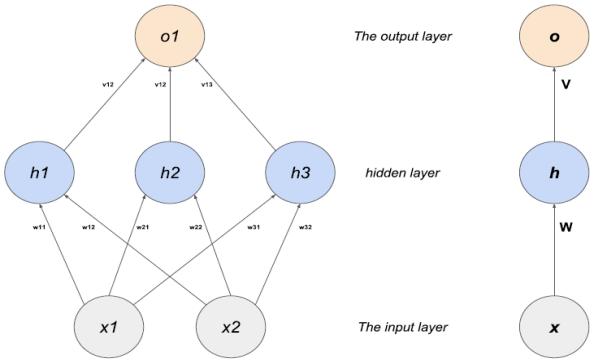

Fig. 4: A feed-forward neural network (FFNN) and its compact form

Figure 4 shows a feed-forward neural network (FFNN) with an input layer of two inputs, a hidden layer of three neurons and a single output layer. The plot in the right shows a compact form of the network. W and V have learned network parameters.

Each layer on an FFNN is formed of a group of *neurons*. Each neuron applies an anon linear transformation to its high dimensional input *I* and produces a single output, in two steps: (1) linearly transforming the input into single output using a *weight matrix W*, and (2) then applying a non-linear transformation *h*. This can be expressed as $h\ (W^T I)$. The weights *W* are the network parameters that will be tuned to create the function approximation. FFNNs has been shown to be a powerful function approximators. In the presence of enough data, the network performance can be increased by increasing the model capacity (by increasing the count of layers and the count of neurons per layer). Yet, FFNNs are not suited for processing sequential data (e.g. text, audio or video), for the below reasons:

- FFNNs use redundant network resources to be able to handle translation on the inputs. As an example, consider training an FFNN to predict the city name from input sentences like ([*Stockholm* is a beautiful city], [I went to *Stockholm*]). The network will have to learn to see the city name on any of its inputs, by using a different set of parameters for each input, instead of sharing a single set of parameters that learned to recognize the city name.
- FFNNs architecture does not explicitly capture the correlation present on the inputs. (X is nice not Y) and (Y is nice not X) look the same for an FFNN, even though the sentences bear different meanings.

4.2.2 Recurrent Neural Network (RNN): Recurrent Neural Networks (RNN) addresses the FFNNs problems mentioned before, by introducing a recurrent connection on its hidden layers as illustrated in Figure 5. At every time step t, the neuron output will be based on not only its current input x_t but also the neuron output from previous time step t−1. This provides a mechanism for the network to capture dependence between correlated input features. Also, since the network parameters are shared amongst all the time steps, RNNs are more efficient than FFNNs. RNNs have successfully been used on tasks involving correlated inputs, like speech recognition, language translation and image and video captioning. RNN computation graphs are typically differentiable and hence trained with gradient descent methods. A loss function, typically an NLL, is defined and minimized (by tuning the network parameters) using the SGD method.

Figure 5 shows a recurrent neural network with one hidden layer. Left is the network diagram. Right is the network computation graph unrolled over time steps *t* −1, *t*, *t* + 1? W, K and V have learned network parameters.

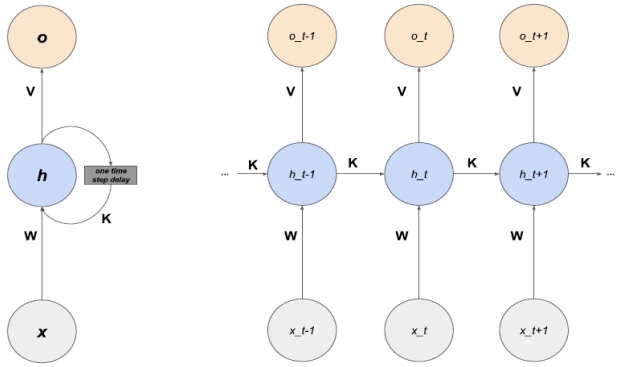

Fig. 5: A Recurrent Neural Network with one hidden layer

4.2.3 Long Short – Term Memory (LSTM): The function composed by RNNs involves the repeated application of the same hidden layer neuron functions. For simplicity, If we excluded the input x and the non-linear transformation h and assumed a scalar hidden to hidden (recurrent connection) weight k, the composed function will appear something like k^τ, where τ is the number of time steps. For large τ values, the product k^τ will vanish (becomes very small) or explode (becomes very big) depending on whether k is smaller or greater than one. And since the gradients calculated for this RNN are scaled by the k^τ product, they will eventually vanish or explode as well. This problem is known as the vanishing and exploding gradient problem.

The vanishing and exploding gradient problem makes training RNNs hard: vanishing gradients result in a very weak signal for the correct parameter update direction that minimizes the loss function (and hence difficulty to learn dependencies overlong sequences), and the exploding gradients makes the training unstable (manifested as rapid big fluctuations in the loss function value).

The exploding gradient problem is commonly solved by clipping the calculated gradients that exceed a threshold value that is learned through cross-validation. *Long Short-Term Memory (LSTM)* cells help solve the vanishing gradient problem. It does that by learning a different weight k_t at each time step, such that the final product $\prod_{t=1}^{\tau} k_t$ neither vanish nor explode.

4.3 Hand Gesture Recognition Model

The RNN model proposed in this paper to predict hand gestures from the Wi-Fi RSSI sequences is an LSTM based RNN model shown in Figure 6 wherein *τ*, *N* and the count of layers are model hyperparameters selected using cross-validation.

The model performance on predicting the correct hand gestures is evaluated using the *accuracy* measure, which can be defined as the percentage of correctly predicted gestures from the total performed test gestures. If the system was tried using a set of gestures with inputs $X = [x1\ x2 \ldots xi \ldots xm]$ and corresponding true labels $Y = [y1\ y2 \ldots yi \ldots ym]$, accuracy is defined as:

$$Accuracy = \frac{1}{m}\sum_{i=1}^{m} I_{yi}(\hat{y}) \quad (8)$$

Where $\hat{y}_i$ is the model prediction for input x_i, and $Iy_i(\hat{y}_i)$ is 1 if $\hat{y}_i = y_i$ and 0 otherwise.

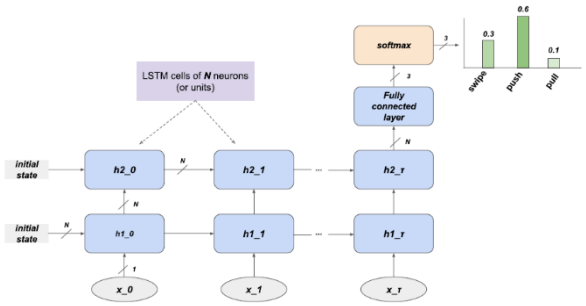

Fig. 6: Time-unrolled diagram of the LSTM RNN model

Besides accuracy, the *confusion matrix* is used to provide a breakdown of the model performance per individual gesture, where each gesture's correct and missed predictions are shown.

5. METHODOLOGY

5.1 Solution description

Figure 7 shows a diagram for the proposed hand gesture recognition solution. The following subsections explain the functional modules of the solution.

5.1.1 RSSI Collection: This sub-module interfaces the wireless device of the smartphone and outputs a stream of RSSI values at a specific rate (~200 values per second in the implemented system).

5.1.2 Windowing: This sub-module splits the incoming RSSI stream into equal length (T) overlapping windows. Both the window length as well as the delay between consecutive windows (d) are specified in seconds. Since the incoming RSSI stream rate is approximate ~ 200 values per second, the output windows from the windowing step will have a variable number of RSSI values per window. Different window sizes T has been investigated in this report. In all the experiments the time delay between successive windows d was set to one second (using other values for the delay d were not investigated).

5.1.3 Noise detection: Only windows with high enough activity, identified by the window variance exceeding a specific threshold, are likely to be caused by hand gestures. Such windows will be forwarded to the subsequent steps of the gesture classification system. All windows that have variance lesser than the threshold will be predicted as no gesture or Noise.

5.1.4 Preprocessing: This sub-module takes as input windows with a variable number of RSSI values per window and outputs windows with an equivalent number of feature values (τ). Each incoming window will be processed as below:

- **Mean subtraction**: In this step, the mean RSSI value of the window is calculated and then subtracted from each of the window's RSSI values. As a result, the window values will be centred around zero. This step increases the system robustness against variations in the RSSI values due to, for example, the increase of RSSI when the phone is moved closer to the AP or the decrease of RSSI when the phone is moved away from the AP.
- **Sampling**: This steps samples τ feature values with a time difference between successive samples equal to T/ τ on average.
- **Standardizing**: Each one of the τ feature values is reduced by the training data mean of that value.
- **Normalizing**: Each one of the τ feature values (standardized in the previous step) is divided by the training data standard deviation of that value.

5.1.5 Interference (LSTM RNN Model): The LSTM RNN model takes input of τ features and outputs three values proportionate to the conditional probability assigned by the model to each possible gesture given the input.

During training, the RNN model outputs are used as inputs to the softmax layer (a layer that computes the softmax function which is a simplification of the logistic function) used to calculate the model loss. Since the softmax layer inputs are known as logits, the RNN model outputs in this solution are called as logits as well.

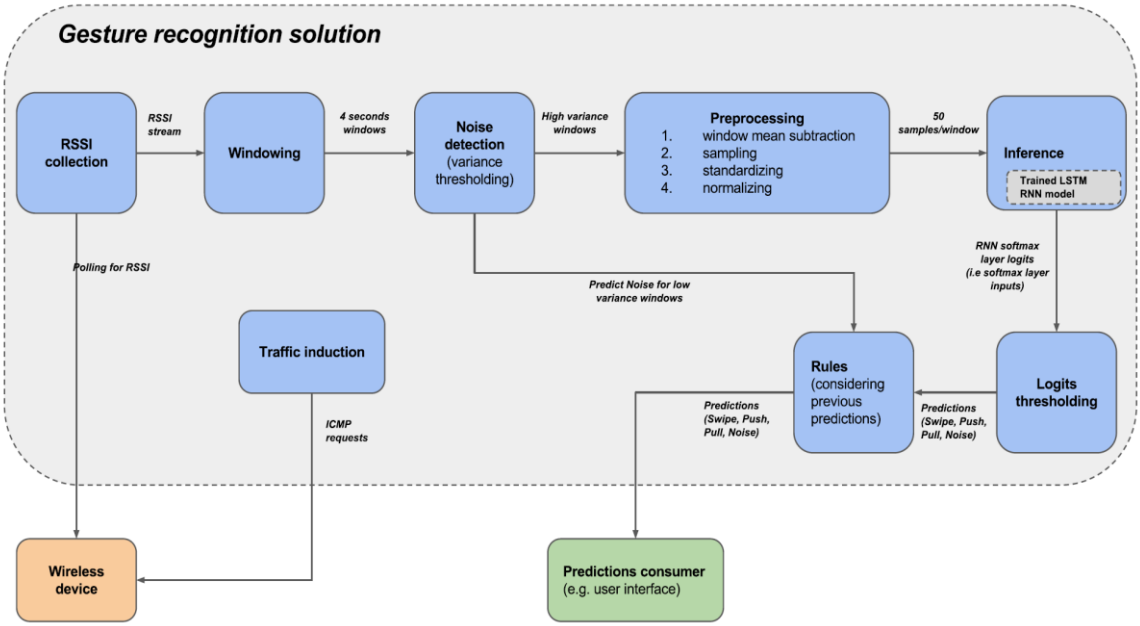

Fig. 7: Gesture Recognition Solution Diagram

5.1.6 Logits Thresholding: This sub-module keeps a short history of the predictions made by the system previously, and applies, set of rules accepting or rejecting the current predictions made by the previous inference and thresholding steps. The rules are:

- Allow Pull gestures after Push gestures only. The Pull gesture signature on the RSSI stream looks like a pause on the RSSI value followed by an increase. This is similar to the increase in the RSSI values resulting from some background activities like when the AP increases its output signal power. This rule decreases the number of false positive Pull predictions produced by such interfering background activities.
- A prediction that is dissimilar to its immediate predecessor is ignored (and Noise is predicted instead). Exempted from this rule are:

(a) Swipe or Push following a Noise prediction
(b) Pull prediction that follows a Push
(c) Noise predictions

The reason for having this rule is because each prediction window overlaps with the previous window (three seconds overlap in most experiments). If the previous window contained a gesture, the following window RSSI stream may look comparable to another gesture than the performed one. For example, the end of Swipe gestures looks comparable to Pull gestures. In all experiments, a gap of two to three seconds is maintained between the gestures since the predictions made by the RNN model during this period (i.e. the two seconds after the previous gesture) will be ignored as implied by this rule.

5.1.7 Traffic Induction: The wireless interface makes new RSS measurement only when a new Wi-Fi frame is received. To ensure that the wireless device makes enough updated RSS measurements, this module induces traffic between the AP and client (smartphone) by sending a constant stream of Internet Control Message Protocol (ICMP) echo requests to the AP. For every ICMP echo request, the AP will transmit an ICMP echo reply back to the smartphone, and the smartphone wireless interface will make an updated RSS measurement.

5.2 System Training

Besides training the LSTM RNN model, training the system also involves deciding the values for the other system parameters (i.e. thresholds). In this project, the training is done in an offline setting.

5.2.1 Offline data preprocessing: For all offline experiments, the steps below were followed to preprocess the collected training data. Note that apart from steps one, four and five, the online recognition solution preprocesses the inward RSSI windows in the way described below.

(a) The RSSI values stream is read from the received data files and then divided into D windows (corresponding to the gestures), every window being T seconds long.
(b) For each window, the average is calculated and then subtracted from the individual window values.
(c) τ values that are equally spaced in time are then sampled from each window. The result is a dataset of shape D windows each having τ features.
(d) The dataset is then randomly split into training (D_{train}= 0.75D) and testing (D_{test}= 0.25D) sets. Furthermore, when a model hyper-parameter selection is done, 0.8D_{train} of the training set is used to train the model, and the remaining D_{val}= 0.20$Dtrain$ is used to select the hyperparameters (validation set).
(e) Using the training set ($D_{train} \times \tau$), the average and standard deviation of each one of the τ features is calculated.
(f) All training and testing set windows were standardized and normalized using the training mean and standard deviation.

5.2.2 LSTM RNN model training: The LSTM RNN model outputs a conditional probability distribution over the possible gestures given the input. The model is trained to minimalize the negative log-likelihood loss, using a variant of SGD known as Adaptive Moment Estimation, or shortly ADAM [30].

Almost all of the model hyperparameters are selected by performing a grid search over the space defined by the hyperparameters. Each parameter set is evaluated using a four folds cross-validation.

5.2.3 Thresholds selection: The variance threshold used by the Noise detection step, is initially estimated as the minimum training data windows variance. This value is then manually optimized to maximize the online prediction accuracy. The same approach is followed to choose the threshold used in the logits thresholding step.

6. IMPLEMENTATION

6.1 Hardware Components used

The setup used for carrying out the different experiments of this paper used the below hardware devices:

(a) A smartphone is used to receive the Wi-Fi RSSI measurements stream while performing hand gestures. The device was also used to evaluate the developed hand gesture recognition solution. The smartphone was a XIOMI Redmi Note 4 running Android 9.0 OS.
(b) One Wi-Fi access points: one operates in both 2.4 GHz and 5 GHz frequency bands, and the other operates in 2.4 GHz frequency band only.
(c) A notebook PC for parsing the collected RSSI data to inputs suitable for training the neural network. It is also used to develop Android applications via Android Studio. The PC had 16 GB of memory and a quad-core Intel processor (Intel(R) Core (TM) i7-4702MQ CPU @ 2.20GHz). It also had a 384 cores Nvidia GPU (GeForce GT 740 M) which was used for training the LSTM RNN model. The PC was running a Mint Linux distribution.
(d) An Arduino Uno board that has the integrated Atmega 328p microcontroller will communicate with the HC-05 Bluetooth module. It also controls a relay that further controls appliances.
(e) A Bluetooth module pairs and connects with the smartphone and receives the ASCII values from the smartphone

6.2 Software tools

6.2.1 Tools used for data collection: To be able to receive the Wi-Fi RSSI measurements while performing the hand gestures and save them for later processing, an Android mobile application was developed using Java. The Android API provided a way for reading the RSSI values measured by the wireless interface, but these values were updated by the Android API at a maximum rate of approximately one time per second. To overcome this limitation, the RSSI measurements are collected directly from the wireless extension for Linux user interface, which is exposed as a pseudo file named /proc/net/wireless. The implementation continuously reads the /proc/net/wireless file and reports the RSSI measurements at a frequency of ~200 values per second.

The Wi-Fi RSSI collection application provides a mean to start, stop and name measurements. It also provides a means to export and deletes saved measurements.

6.2.2 Tools used for offline analysis: The offline analysis phase involves exploring the collected RSSI data and evaluating and tuning a set of dissimilar classification algorithms. Python was the main language used in this phase, and that is due to:

- Python has a set of powerful and important libraries for data and signal processing like *numpy*, *pandas and scipy*.
- The abundance of off-the-shelf machine learning algorithm implementations.
- Many Python-based machine learning frameworks provide support for exporting trained models to be used on other setups. For instance, Tensorflow provides a way to export a trained model, and then utilize the exported model on a mobile application.

The Tensorflow framework was used to build and train the LSTM RNN model. The implementation of most of the other machine learning algorithms evaluated in this thesis was mainly provided by the *UEA and UCR Time Series Classification Repository*, except the *LTS Shapelets* and *DTW-KNN* which were implemented in Python as part of this paper.

6.2.3 Tool used for microcontroller programming: The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written using the Java programming language. It is used to write and upload codes to Arduino compatible boards. In the program code, we include the Bluetooth module's operation specifications, such that the reception of ASCII signal from the smartphone is received appropriately. Also, another program was input into the Arduino board and according to this program; the Arduino board controls the relay.

7. EXPERIMENTS

7.1 Performed Hand Gestures

Three hand gestures are considered. In all performed experiments, the smartphone was placed on a flat surface table.

- **Swipe gesture**: it involves moving the hand above the smartphone (around five centimetres above the phone) from one side to the other and back to the starting position.
- **Push gesture**: here the hand is moved down towards the smartphone and placed steadily above it (around five centimetres) for about two seconds.
- **Pull gesture**: it involves placing the hand above the smartphone (around five centimetres) steadily for about two seconds before moving it upward.
- Notice that, the gesture recognition solution allows Pull gestures after Push gestures only.

7.2 Data Collection

A dataset was collected to train the LSTM RNN model as well as to tune the different recognition system parameters.

7.2.1 Traffic scenarios: Three traffic scenarios between the AP and the smartphone were considered when collecting the data:

- (Internet access + traffic induction): in this scenario, the AP is connected to the internet and hence the smartphone (via the AP). At the same time, the smartphone is continuously sending ICMP requests to the AP (pinging the AP) at a rate of ~700 times/second.
- (No internet access + traffic induction): both the AP and the smartphone does not have internet access, but the smartphone is uninterruptedly pinging the AP at a frequency of ~700 times/second.
- (No internet access + no traffic induction): the smartphone has neither internet access (via the AP), nor does it ping the AP.

7.2.2 Collection Procedure: The dataset was received during times with minimal human activity (i.e. walking), to reduce the interfering noise introduced by such activities in the Wi-Fi signal. This also made it difficult to have two or more subjects to perform gestures.

A mobile application was developed specifically for recording the Wi-Fi RSSI data. The application records the RSSI provided by the smartphone wireless interface at a frequency of ~200 samples/second. A typical collection session commenced as below:

(a) The smartphone is connected to the AP.
(b) The RSSI collection application is initiated.
(c) At a specific point in time (start time), the subject starts the performance of the gestures. Successive gestures are separated by a ten seconds gap (gap time). Both the start and gap times are noted and used later to obtain the gesture windows (parts of the received RSSI stream that match the hand gestures).
(d) The received RSSI stream is kept in the phone as a text file with a name specifying the performed gesture. The file is then transferred to a PC.

7.2.3 Gesture Windows Extraction

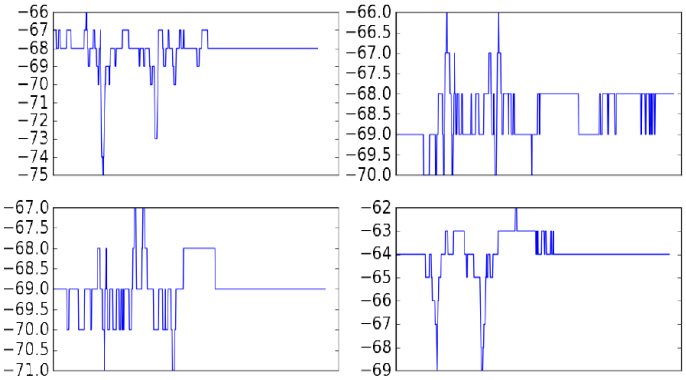

Fig. 8: Four seconds window of Swipe Gesture

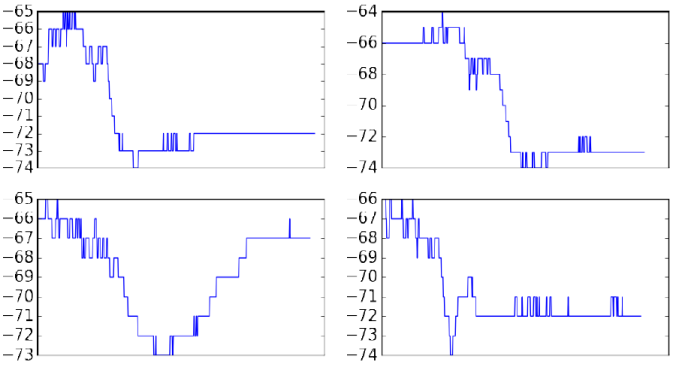

Fig. 9: Four seconds window of Push Gesture

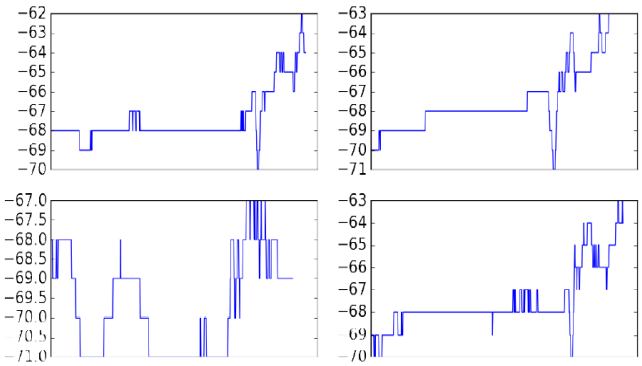

Fig. 10: Four seconds window of Pull Gesture

To successfully train the recognition model, the correct RSSI windows which correspond to the gestures have to be used, and hence extracted from the collected RSSI stream. The window extraction is done using the start and gap times values introduced in the previous section. Figures 8, 9 and 10 show sample windows for Swipe, Push and Pull gestures respectively. The reported mean accuracies in offline experiments are calculated by evaluating the RNN model ten times on the specific configurations being tested, each using a different random split of the data into testing and training sets.

7.3 LSTM RNN model training and evaluation

The model is trained with an SGD variant known as ADAM. Unless explicitly specified otherwise, the model parameters used on all offline and online experiments are shown in Table 1.

Table 1: LSTM RNN model parameters and hyper parameters

parameter	value
RNN time steps (T)	50
Number of hidden LSTM layers	2
Number of units (or neurons) per LSTM (N)	200
Learning rate	0.001
SGD batch size	50
Dropout probability	0.5
Model parameters initial random values boundaries	±0.08
Maximum gradient norm (for clipping big gradients)	25
Number of training iterations	600

8. CONCLUSION

The work in this paper demonstrated that it is possible to identify and categorize contact-less hand gestures in motion near smartphones without modification to the smartphone components (hardware) or software. The implemented solution uses an LSTM RNN model to predict the performed hand gesture from the smartphone Wi-Fi RSSI stream. The solution achieved average recognition accuracy of 78% when tested on several online scenarios including ones that are different from the scenarios under which the system was trained. This accuracy qualifies the gesture recognition solution for non-mission critical mobile applications. The main limitations of the developed solution are its vulnerability to interfering background activities and its power consumption. These limitations are addressed by the preamble detection mode which reduces the false positive prediction rate and reduces the system power consumption.

9. ACKNOWLEDGMENT

We would like to thank Dr Ambedkar Institute of Technology, India for its invaluable support.

10. REFERENCES

[1] N. Duta. A survey of biometric technology based on hand shape. Pattern Recognition 42 (11), 2009, pp. 2797-2806.
[2] K. Karu, and A.K. Jain. 1996. Fingerprint classification. Pattern Recognition 29 (3), pp. 389-404.
[3] D. Malaspina, E. Coleman, R.R. Goetz, J, Harkavy-Friedman, C. Corcoran, X. Amador, S. Yale, and J.M. Gorman. Odor identification, eye tracking and deficit syndrome schizophrenia. Biological Psychiatry 51 (10), 2002, pp. 809-815.
[4] H.A. Park, and K.R. Park. Iris recognition based on score level fusion by using SVM. Pattern Recognition Letters 28 (15), 2007, pp. 2019-2028.
[5] K Kurita. Human Identification from Walking Signal Based on Measurement of Current Generated by Electrostatic Induction. In Proceedings of the 2011 International Conference on Biometrics and Kansei Engineering (ICBAKE '11), 2011, pp. 232-237.
[6] JJ Little, and JE Boyd. Recognizing People by Their Gait: The Shape of Motion. Journal of Computer Vision Research 1(2): 2232. 40244.
[7] C.Nickel, C.Busch, S.Rangarajan, and M.Möbius. Using Hidden Markov Models for accelerometer-based biometric gait recognition. Signal Processing and its Applications (CSPA), 2011 IEEE 7th International Colloquium on, 2011, pp. 58-63.
[8] D. Mulyono, and H.S. Jinn. A study is of finger vein biometric for personal identification. International Symposium on Biometrics and Security Technologies, 2008, pp. 1-8.
[9] "Google Project Soli," https://www.google.comJatap/project-soli/.
[10] R. Chellappa, C. Wilson, and S. Sirohev. Human and machine recognition of faces: a survey. In Proceedings of IEEE vol. 83 (5), 1995, pp. 705-740.
[11] Air Gesture on Samsung S4 works under well-lit conditions. http://touchlessgeneration.com/discover-touchless/testing-of-air-gestures-on-the-galaxy s4/#.UkSVWIY3uSo.
[12] Gupta, S., Morris, D., Patel, S., and Tan, D. Soundwave: using the Doppler effect to sense gestures. In HCI (2012).
[13] Kinect. https://dev.windows.com/en-us/kinect.
[14] Leap Motion. https://www.leapmotion.com/.
[15] J. M. Rehg and T. Kanade. Visual tracking of high doff articulated structures: an application to human hand tracking. In Springer Computer Vision–ECCV. 1994.
[16] S. Agrawal, I. Constandache, S. Gaonkar, R. Roy Choudhury, K. Caves, and F. DeRuyter.Using mobile phones to write in the air. In ACM MobiSys, 2011.
[17] J. Wang, D. Vasisht, and D. Katabi.Rf-idraw: virtual touch screen in the air using rf signals. In ACM SIGCOMM, 2014.
[18] A. Parate, M.-C. Chiu, et al. Risq: Recognizing smoking gestures with inertial sensors on a wristband. In ACM MobiSys, 2014.
[19] Q. Pu, S. Gupta, S. Gollakota, and S. Patel.Whole-home gesture recognition using wireless signals.In ACM MobiCom, 2013.
[20] Y. Ren, C. Wang, Y. Chen, M. C. Chuah, and J. Yang. Critical segment based real-time e-signature for securing mobile transactions. In IEEE CNS, 2015.
[21] H. T. Friis. A note on a simple transmission formula. Proceedings of the IRE, 34(5):254–256, May 1946.
[22] Abdollah Ghasemi, Ali Abedi, and Farshid Ghasemi. Basic Principles in Radiowave Propagation, pages 23–55. Springer New York, New York, NY, 2012.
[23] Ian Poole. Electromagnetic waves and radio propagation. http://www.radio-electronics.com/info/propagation/em_waves/electromagnetic_waves.php. [Online; accessed 16-May-2016].
[24] Rajalakshmi Nandakumar, Bryce Kellogg, and Shyamnath Gollakota. Wi-fi gesture recognition on existing devices. CoRR, abs/1411.5394, 2014.
[25] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. Tool release: Gathering 802.11n traces with channel state information. SIGCOMM Comput. Commun. Rev., 41(1):53–53, January 2011.
[26] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. Tool release: Gathering 802.11n traces with channel state information. ACM SIGCOMM Computer Communication Review (CCR), January 2011, 2011.
[27] Stephan Sigg, Mario Hock, Markus Scholz, Gerhard Tröster, Lars Wolf, Yusheng Ji, and Michael Beigl. Mobile

and Ubiquitous Systems: Computing, Networking, and Services: 10th International Conference, MOBIQUITOUS 2013, Tokyo, Japan, December 2-4, 2013, Revised Selected Papers, chapter Passive, Device-Free Recognition on Your Mobile Phone: Tools, Features and a Case Study, pages 435–446. Springer International Publishing, Cham, 2014.

