A CSI-based Position Independent Gesture Recognition System to Operate Smart Home Appliances

Presented By
Zerin Shaima Meem
ID: 1804057
Dept. of CSE, CUET

Supervised By
Dr. Mahfuzulhoq Chowdhury
Professor
Dept. of CSE, CUET

Course Code: CSE:400
Course Title: Project & Thesis

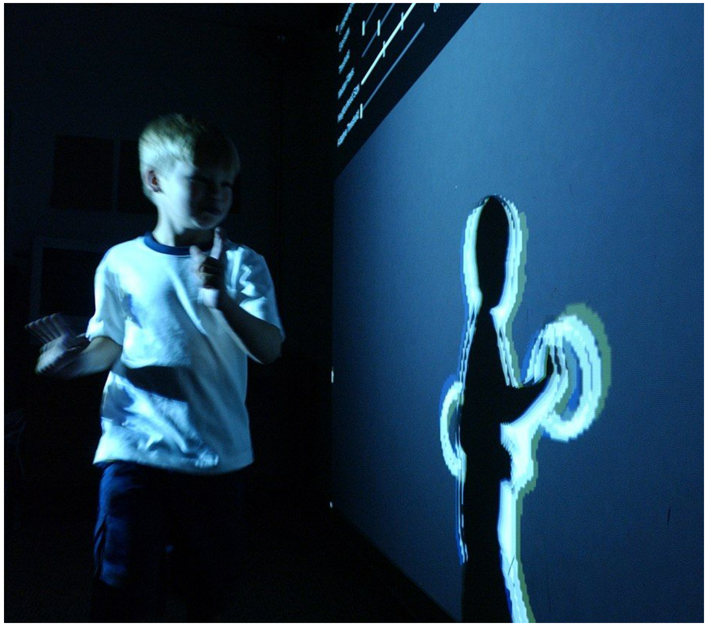
Contents

Introduction

Gesture recognition is a technology that enables computers to interpret and respond to human gestures as inputs.

Widely used gesture recognition systems:

1. Computer Vision based
2. Wearable Sensor based
3. WiFi Sensing based
   - ★ RSSI (Coarse grained)
   - ★ CSI (Fine grained)



Introduction (contd…)

Among all other methods, CSI-based systems gained attention, because they:

- are privacy-preserving,
- have non-intrusive recognition capabilities,
- utilize existing Wi-Fi infrastructure,
- provide fine grained information, and
- are robust against changes in environment.

Introduction (contd…)

- CSI provides detailed channel frequency response (CFR) in OFDM form. This can be represented as:

$$Y(f, t) = H(f, t) \times X(f, t) + N$$

Here,

- $Y(f,t)$ = Received Signal
- $X(f,t)$ = Transmitted Signal, &

$$H(f, t) = \|H(f, t)\| \times \exp^{i\angle H(f,t)}$$

  - Where, $\|H(f, t)\|$ = Magnitude
  - and $\angle H(f, t)$ = Phase
  - of received signal of subcarrier f at time t.

Motivation

IoT applications: Operating smart home appliances, through gestures.

- **Camera-based methods:**
  - sufficient illumination, privacy concerns
- **Wearable Sensor-based methods:**
  - expensive, inconvenient
- **WiFi Sensing-based methods:**
  - position dependent, carry extra device

Related Work

[1] Tracking On-Desk Gestures Based on Wi-Fi CSI on Low-Cost Microcontroller

(Marwa et al., IEEE-ICMU 2023)

- Used ESP32 toolkit for collecting CSI data
- Showed a comparison between SVM, RF, NB and GBC algorithm for evaluating CSI data.
- Number of Gesture Class: 7
- Highest accuracy 72% with Random Forest model.

Limitation:

- User must be on his desk for gesture recognition.
- Didn’t utilize any deep learning algorithm.

Related Work (contd…)

[2] Wi-fi csi based human sign language recognition using lstm network

(Hasmath et al., IEEE-IAICT 2021)

- Used Intel 5300 NIC for collecting CSI data
- Compared the influence of using (i) amplitude values and (ii) amplitude and phase values together as features.
- Number of Gesture Class: 6
- Showed that LSTM outperforms CNN, with 75-78% recognition rate.

Limitation:

- Not position independent.
- Low recognition rate.

Related Work (contd…)

*[3] **Preliminary Investigation of Position Independent Gesture Recognition Using Wi-Fi CSI***

(K. Ohara et al., IEEE-PerCom Workshops 2018)

- Used Intel 5300 Wi-Fi NIC for collecting CSI data.
- CSI obtained from a smartphone carried by a user.
- Number of Gesture Class: 5
- Used Hidden Markov Models (HMMs) and got 67.8% accuracy.

Limitation:

- User always need to carry a smartphone.

Research Question

How to develop a CSI based, device free position independent gesture recognition system with high accuracy?

Objectives

- To **design** a position-independent gesture recognition system for practical uses like operating smart home appliances.
- To **classify** CSI data of four different gestures, from diverse positions in the room, using different machine learning and deep learning models.
- To **compare** the performance of machine learning models like Random Forest, XGBoost, LGBM, and Shapelet Learning; and Deep Learning models like CNN and LSTM; for these types of sequential data.

Applications

- ❑ Smart Home Automation
- ❑ Healthcare Monitoring
- ❑ Human-Computer Interaction (HCI)
- ❑ Security Systems
- ❑ Virtual and Augmented Reality (VR/AR)
- ❑ Automotive Interfaces

Challenges

- ❑ Configuring ESP32 Microcontrollers
- ❑ Establishing a Controlled Environment
- ❑ Preprocessing Raw Data
- ❑ Parsing CSI Data
- ❑ Dataset Labeling and Reshaping
- ❑ Training Models

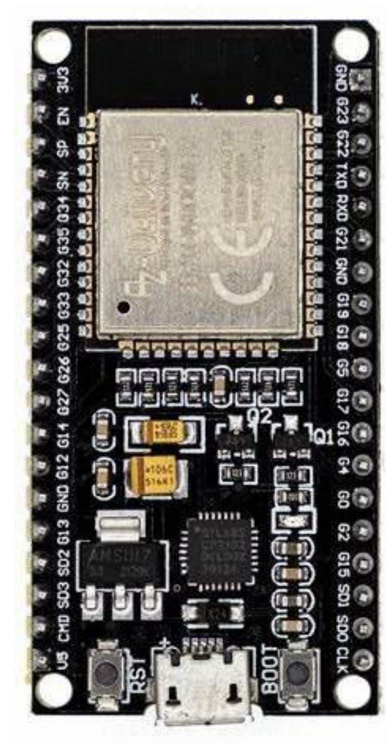


Figure 1: ESP-32 Microcontroller

Contributions

- Using single pair of ESP-32 micro-controllers, for CSI data collection.
- **Collecting 360 sets of training data from 9 fixed points, & 20 sets of testing data from random points with a total of 2,63,660 data packets.**
- Comparing ensemble learning methods: XGB, LGBM, and Random Forest.
- Implementing a Shapelet-Learning method.
- Comparing deep learning methods: CNN & LSTM
- Evaluating all these ML & DL techniques mentioned above.
- Proposing a position-independent gesture recognition system.

Proposed Methodology

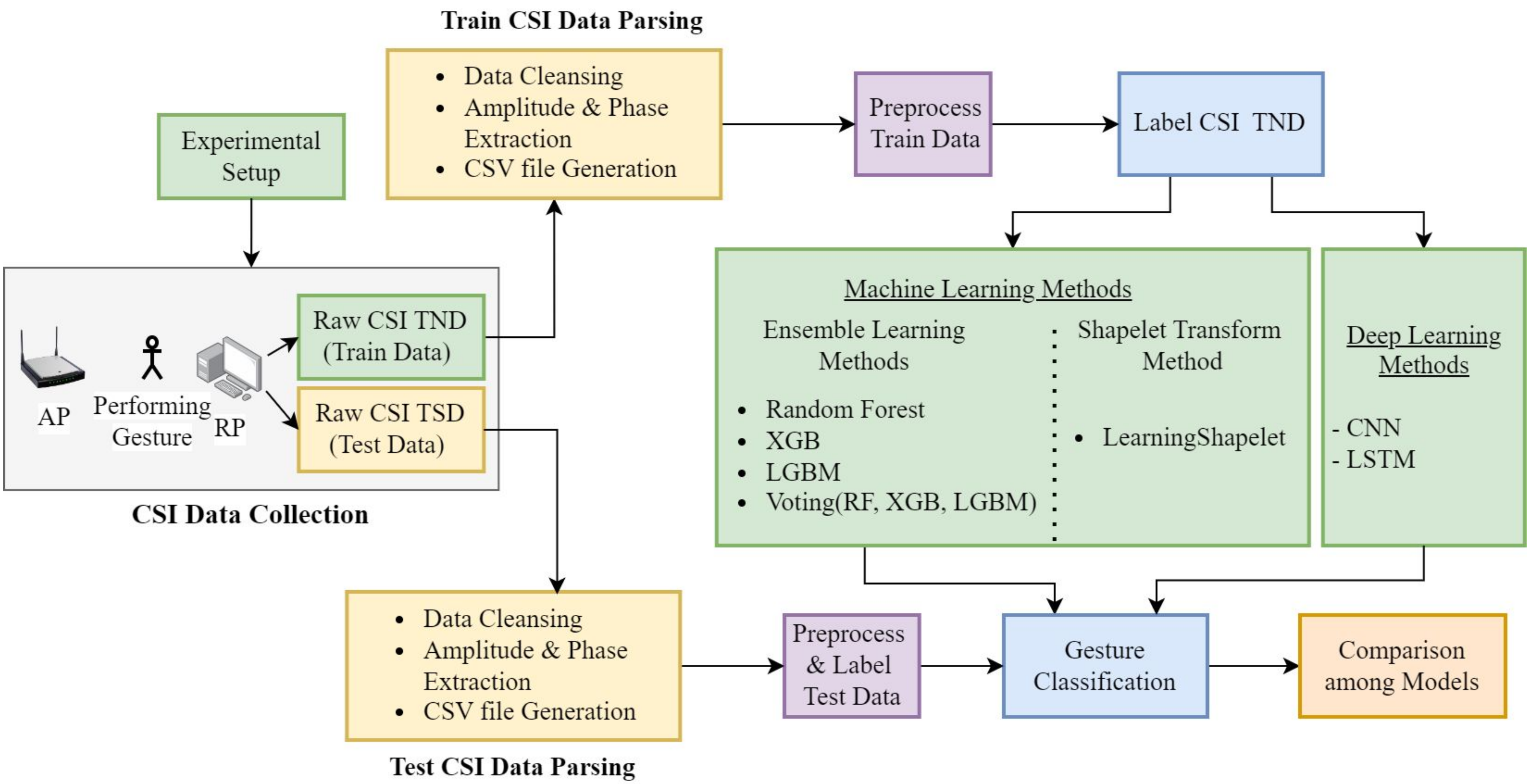


Figure 2: Overview of proposed system

Hardware Configuration

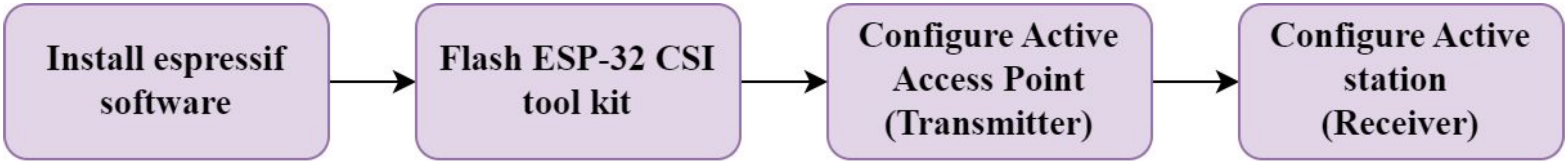


Figure 5: Hardware Configuration Steps

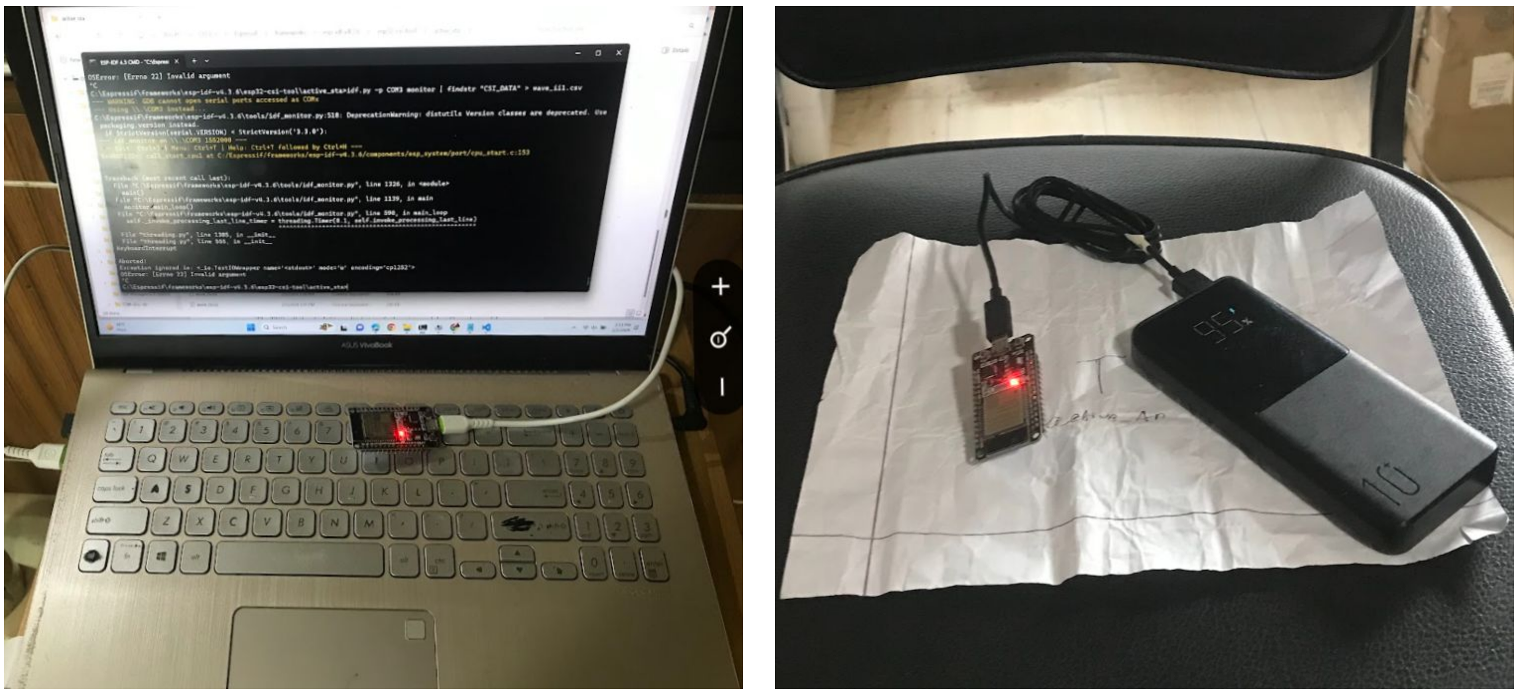


Figure 6: Hardware Setup

Room Arrangement

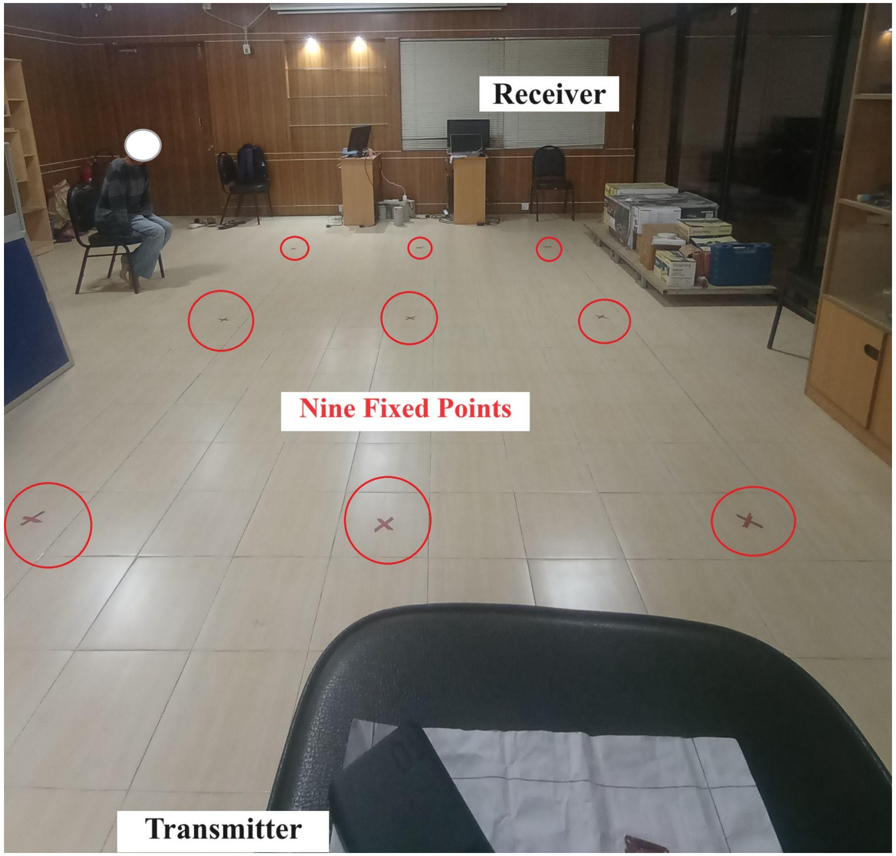


Figure 7: Marked Fixed Points

Figure 8: Actor ready to give Gesture

CSI Data Collection

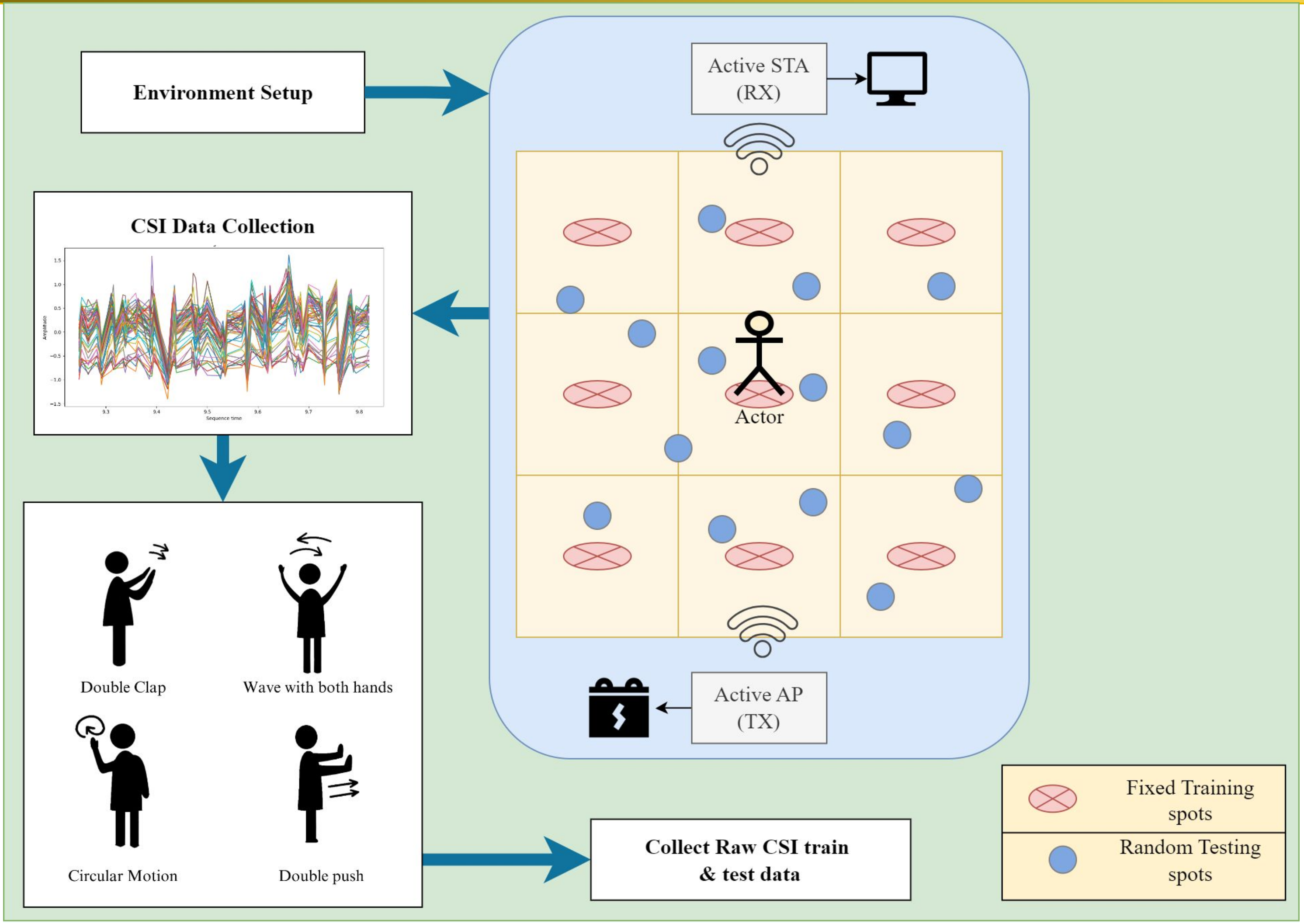


Figure 9: Overview of Data Collection

CSI Data Collection(contd…)

```
wifi not connected. waiting...
CSI_DATA,STA,70:4C:A5:F5:B3:D3,-73,11,0,0,0,0,0,0,0,0,0,-96,0,11,0,8505888,0,269,0,0,8.67545,128,[13 -47 16 0 23 20 17 17 21 18 20 17 25 17 22 15 19 15 17 1
7 15 17 14 17 11 15 13 18 9 15 4 15 0 16 0 15 -3 14 -4 13 -7 11 -9 8 -10 8 -12 7 -10 5 -11 3 -13 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5 10 2 11 1 1
2 -1 13 -1 14 -1 14 -1 16 -1 15 -1 17 -1 17 -1 21 1 18 3 20 7 21 5 18 9 20 10 19 12 21 14 19 17 21 21 20 23 21 21 19 21 17 23 21 23 18 ]
CSI_DATA,STA,70:4C:A5:D3:8D:E8,-77,11,0,0,0,0,0,0,0,0,0,-96,0,11,0,8516697,0,271,0,0,8.68586,128,[15 -15 16 0 3 -15 2 -15 5 -10 5 -10 6 -8 7 -6 9 -6 7 -6 7
-5 9 -6 4 -5 3 -5 -2 -6 -6 -6 -7 -8 -12 -7 -15 -9 -17 -13 -23 -14 -24 -17 -26 -19 -28 -19 -30 -18 -30 -17 -26 -17 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 -18 15 -17 10 -13 5 -10 -1 -6 -5 -1 -10 4 -14 8 -21 13 -23 15 -25 16 -26 20 -29 19 -29 20 -29 20 -29 19 -30 15 -31 15 -28 14 -27 11 -27 7 -25 6 -23 3 -22
0 -20 2 -19 -2 -18 ]
CSI_DATA,STA,70:4C:A5:F5:B1:F0,-89,11,0,0,0,0,0,0,0,0,0,-96,0,11,0,8524513,0,271,0,0,8.69364,128,[15 -15 16 0 11 4 11 0 12 7 15 10 15 7 15 4 8 9 16 3 14 3 1
4 8 13 6 9 3 9 4 6 3 9 4 4 13 4 5 6 4 6 8 7 4 0 3 -3 2 -1 1 7 -1 4 -6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 -11 4 -12 2 -8 6 -11 8 -7 12 -12 8 -11 8
 -6 3 -6 7 -2 9 -4 10 -11 12 -2 4 -8 7 0 11 -13 9 1 4 -7 0 -3 1 -1 9 1 8 -6 5 0 3 -2 6 5 8 4 ]
CSI_DATA,STA,70:4C:A5:F5:89:10,-76,11,0,0,0,0,0,0,0,0,0,-96,0,11,0,8544125,0,271,0,0,8.71328,128,[15 -15 16 0 4 8 6 12 5 12 6 13 8 19 5 16 6 14 5 16 5 17 3
17 4 14 4 14 2 14 0 7 0 8 -3 8 -1 5 -4 7 -5 5 -9 3 -11 1 -13 2 -13 -3 -16 -2 -18 -6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 12 -23 12 -24 15 -24 15 -28
17 -28 16 -29 17 -33 19 -34 18 -33 15 -34 16 -34 15 -35 13 -31 13 -31 12 -32 9 -30 6 -30 6 -26 6 -22 6 -21 4 -18 4 -15 4 -11 2 -10 5 -5 3 -2 ]
CSI_DATA,STA,70:4C:A5:D3:84:41,-69,11,0,0,0,0,0,0,0,0,0,-96,0,11,0,8544835,0,264,0,0,8.71754,128,[8 -127 16 0 8 -11 10 -12 7 -12 8 -16 12 -18 14 -20 7 -23 1
4 -20 11 -23 16 -26 22 -23 6 -28 11 -26 2 -20 20 -19 13 -18 7 -17 -5 -13 -9 -12 13 -19 1 -11 -7 -4 -15 -16 -2 -21 -7 -8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 10 -3 12 -3 11 -3 12 -2 12 -1 12 -1 10 1 10 0 9 2 7 0 6 2 6 0 4 1 3 0 2 0 1 0 0 -1 0 -2 0 -3 1 -3 0 -4 1 -5 2 -5 3 -4 5 -7 6 -8 ]


C:\Espressif\frameworks\esp-idf-v4.3.6\esp32-csi-tool\active_sta>idf.py -p COM8 monitor | findstr "CSI_DATA" > test1.csv
--- WARNING: GDB cannot open serial ports accessed as COMx
--- Using \\.\COM8 instead...
C:\Espressif\frameworks\esp-idf-v4.3.6\tools/idf_monitor.py:518: DeprecationWarning: distutils Version classes are deprecated. Use packaging.version instead
.
  if StrictVersion(serial.VERSION) < StrictVersion('3.3.0'):
```

Figure 10: CSI Data Receiving from Transmitter

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	type	role	mac	rssi	rate	sig_mode	mcs	bandwidth	smoothing	not_sounding	aggregation	stbc	fec_coding
2	CSI_DATA	STA	30:AE:A4:9C:A5	-67	11	0	0	0	0	0	0	0	0
3	CSI_DATA	STA	30:AE:A4:9C:A5	-67	11	0	0	0	0	0	0	0	0
4	CSI_DATA	STA	30:AE:A4:9C:A5	-74	11	1	3	1	1	1	0	0	0

S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE
local_timestamp	ant	sig_len	rx_state	real_time_set	real_timestamp	len	CSI_DATA					
23753	0	34	0	0	0.211462	128	[34 32 2 0 0 0 0 0 0 0 0 0 22 -21 23 -19 26 -18 24 -18 24 -17 25 -15 26 -14 23 -14 22 -12 22 -12 24 -12 21 -13					
37148	0	131	0	0	0.224475	128	[-125 48 8 0 0 0 0 0 0 0 0 0 6 28 5 27 2 30 2 30 0 30 -2 29 -2 29 -4 26 -3 26 -4 26 -4 26 -3 24 -2 24 -2 23 -2 23					
41305	0	135	0	0	0.228841	384	[-121 112 8 0 0 0 0 0 0 0 0 0 7 -25 7 -24 10 -22 12 -25 14 -21 12 -23 11 -23 11 -20 16 -17 15 -21 17 -17 15 -18					

Figure 11: CSI Data in csv. Format

Data Preprocessing

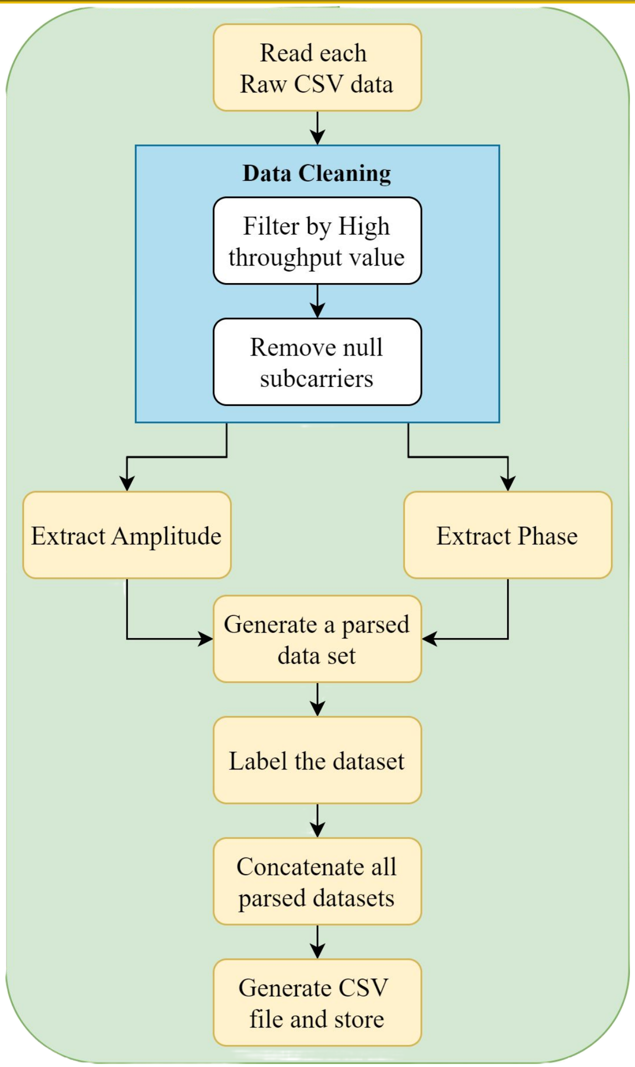
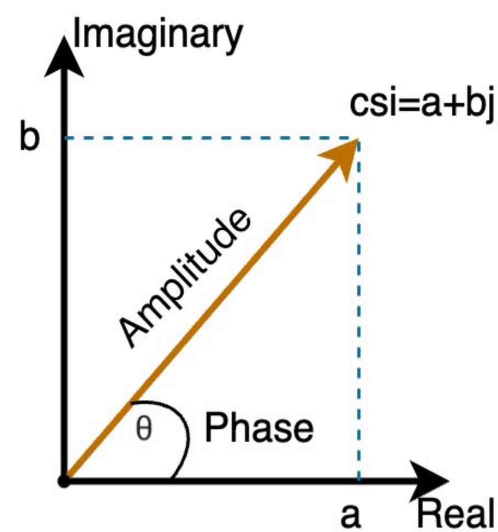


Figure 12: Data Preprocessing

- Sig_mod=1 indicates high throughput
- Amplitude & Phase Extract:



$$Amplitude = \sqrt{a^2 + b^2}$$

$$Phase = \tan^{-1}\frac{b}{a}$$

CSI Data Analysis(contd…)

- Amplitude and Phase data, before and after filtering

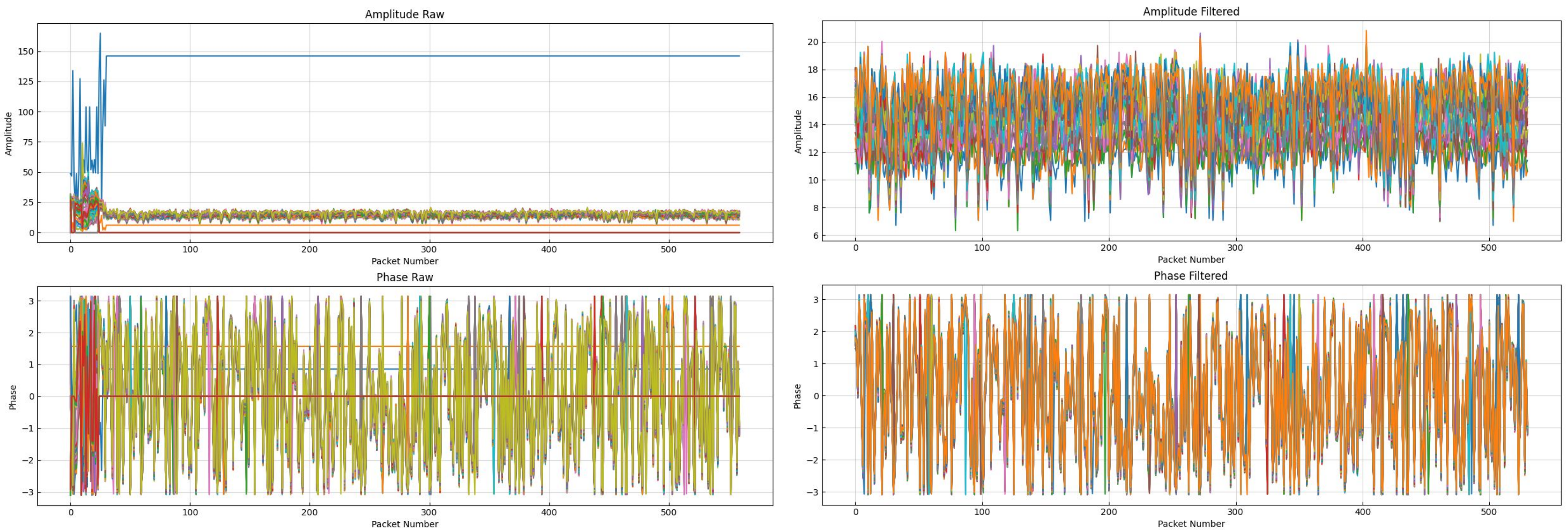


Figure 15: Raw vs Filtered data

CSI Data Analysis(contd…)

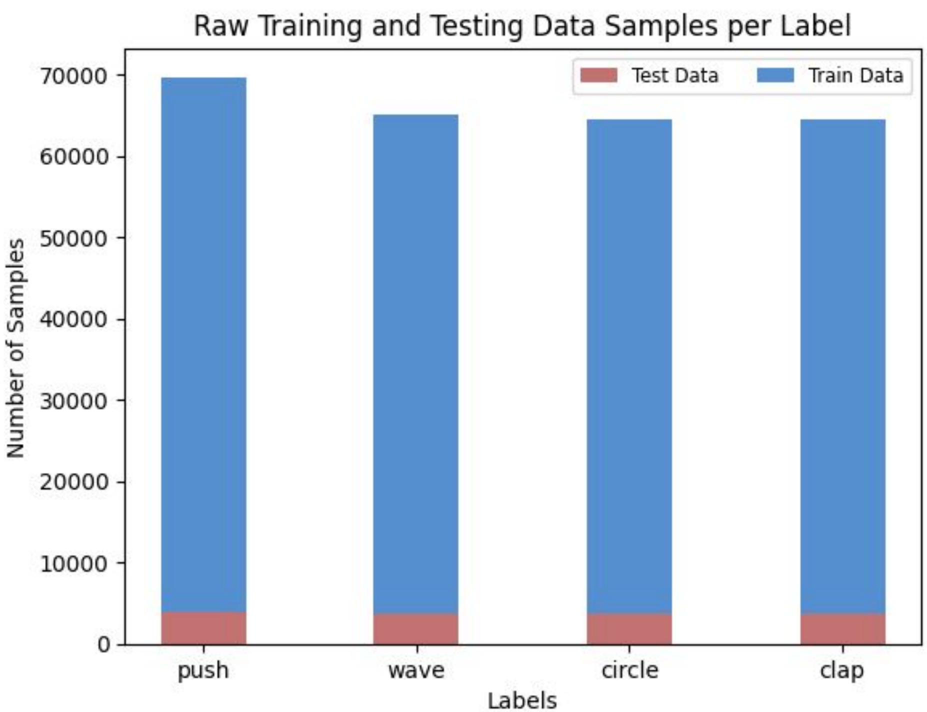


Figure 16(a): Before Filtering

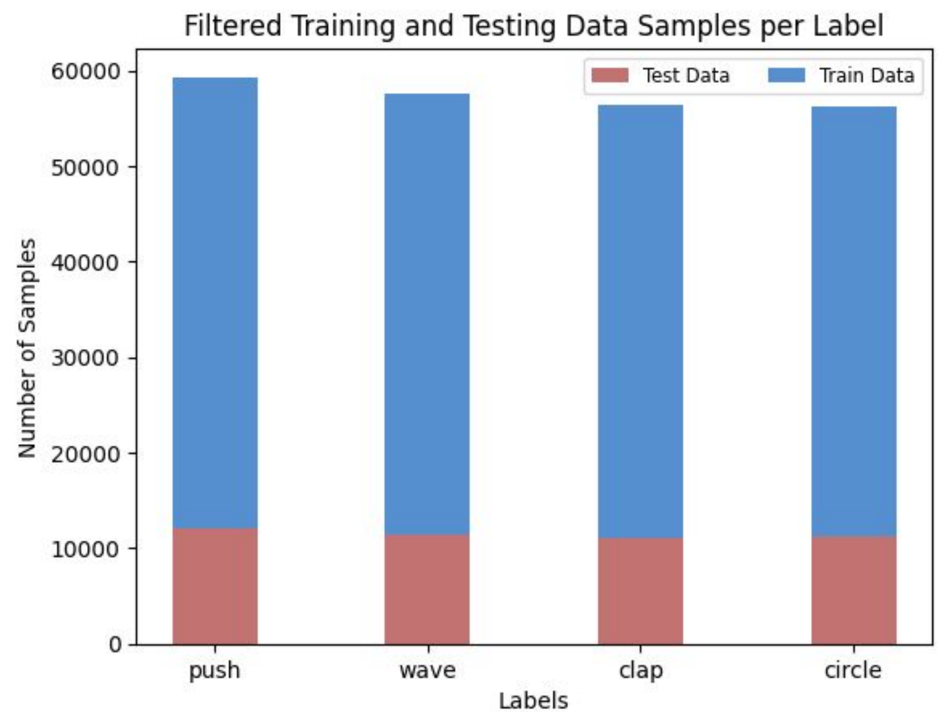


Figure 16(b): After Filtering

Figure 16: Data Packets per Class

2,63,660
Raw Data Packets

2,29,490
Filtered Data Packets

Approached Models

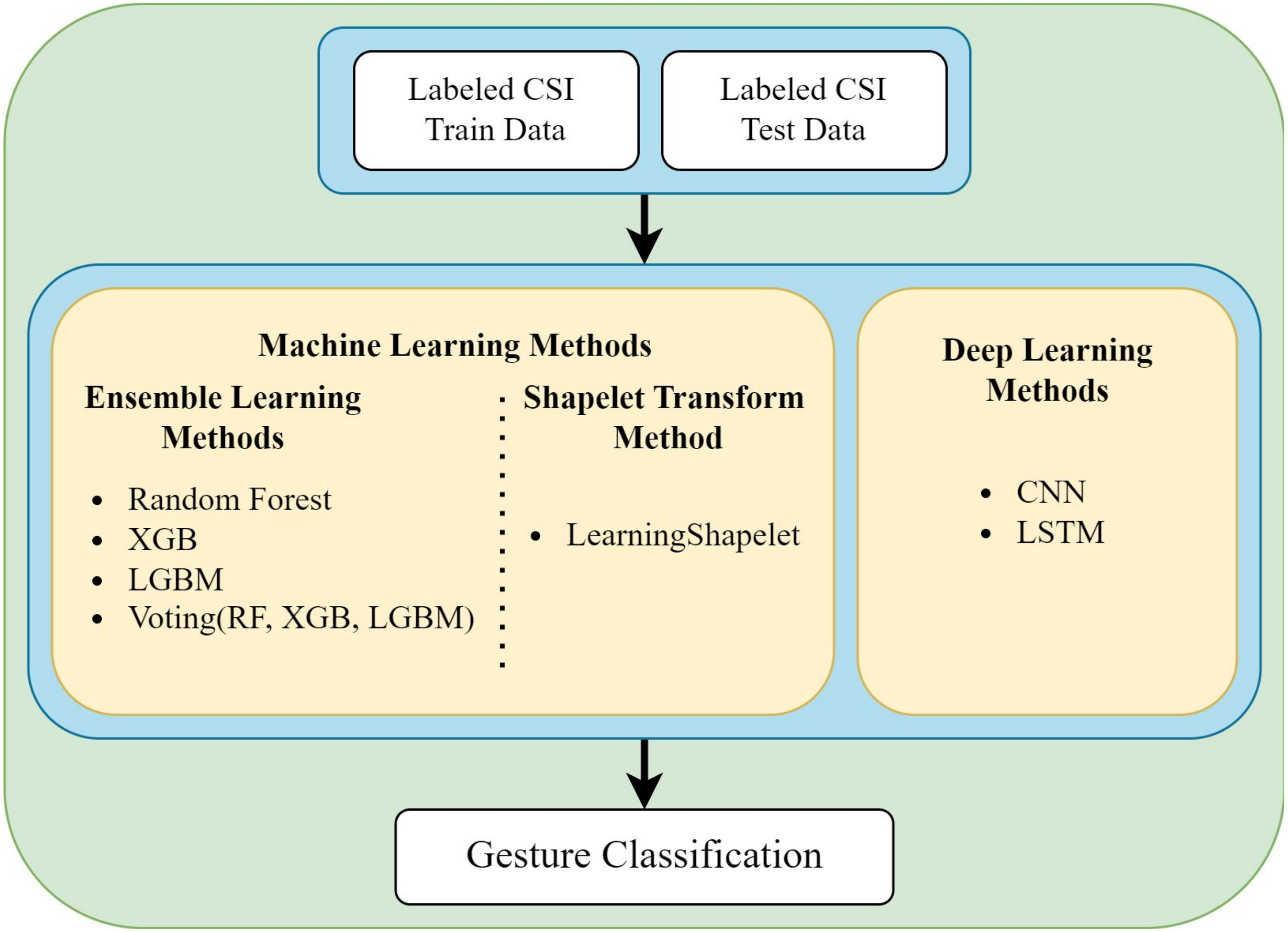


Figure 18: Approached Models

Feature Importance

- Amplitude sub-carriers have higher features importance than Phase sub-carriers.

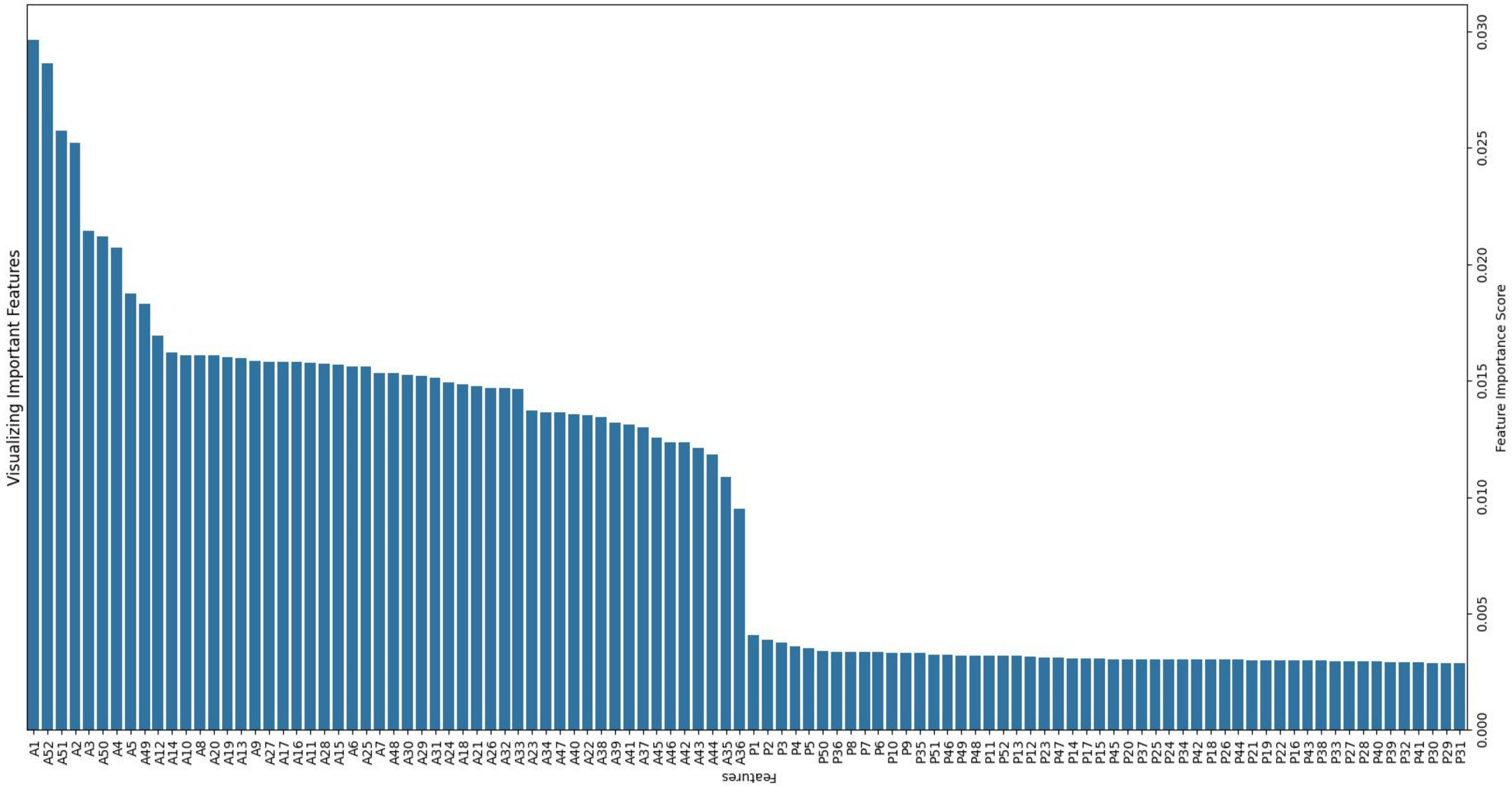


Figure 19:Feature Importance

Ensemble Learning Methods

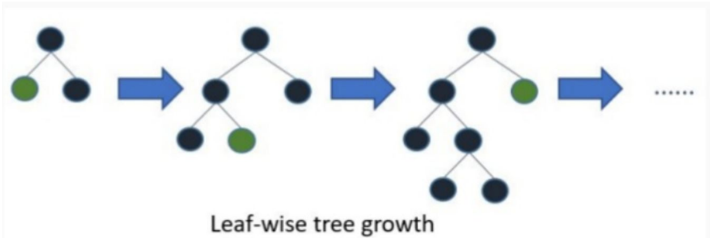


Figure 20: Light-GBM

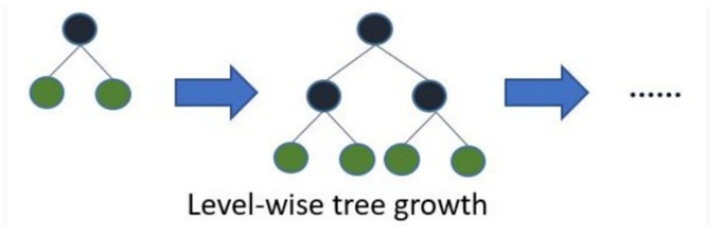


Figure 21: XGBoost

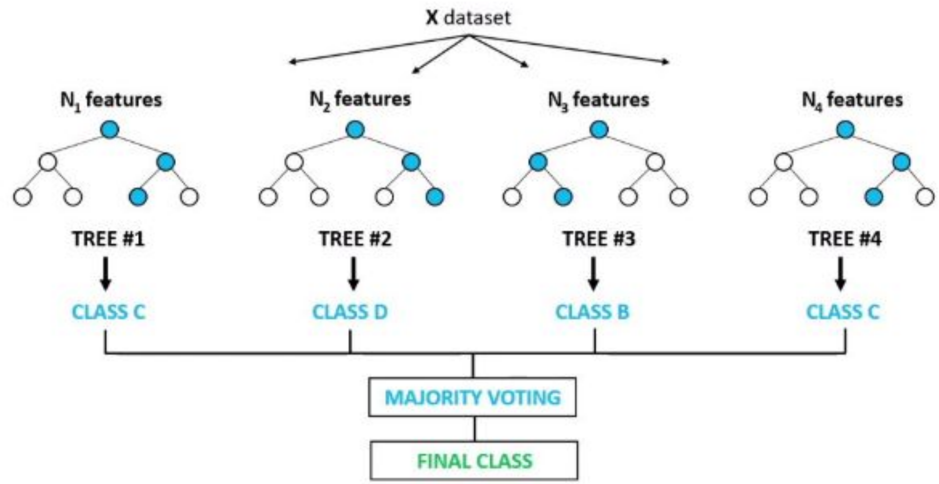


Figure 22: Random Forest

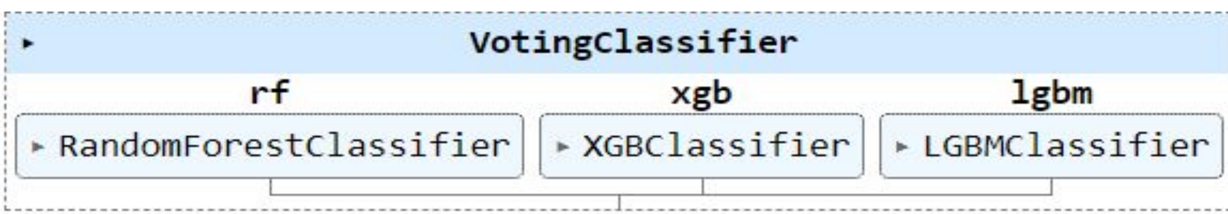


Figure 23: Soft-Voting Ensemble Learning Method

Shapelet Learning Method

- Shapelet is a short segment of a time series.
- CSI data is a time series data.

Shapelet Transformation

- Transferred our dataset from 2D to 3D
- Each instance of transformed data is a time series of window size 200.
- Implemented LearningShapelet model, proposed by J. Grabocka et al. in[5].
- Calculated top-K near to optimal shapelets.

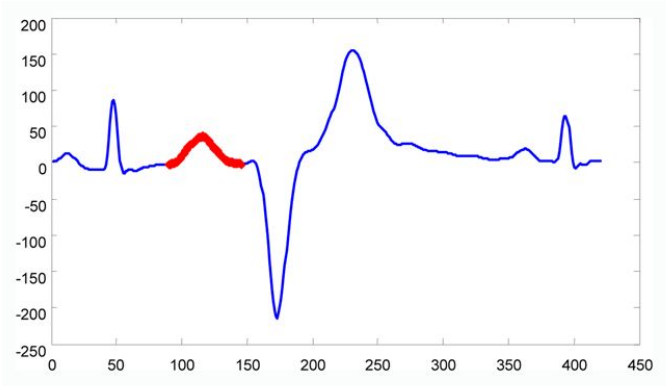


Figure 24: Shapelet

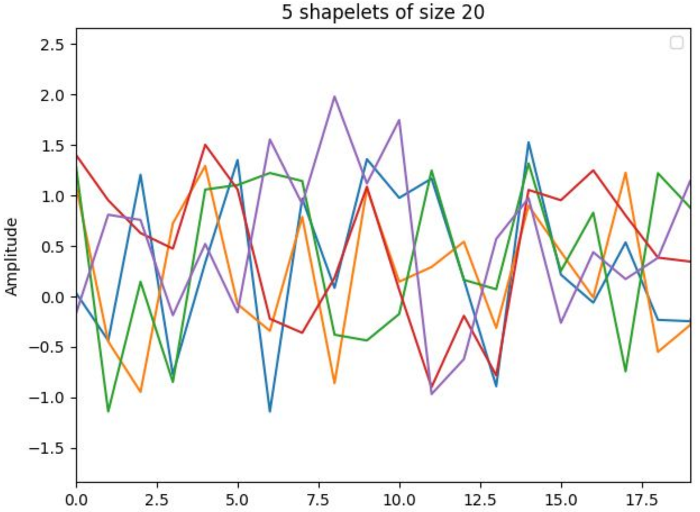


Figure 25: Top 5 Shapelets

Deep Learning Approach: CNN

1D Convolutional Network

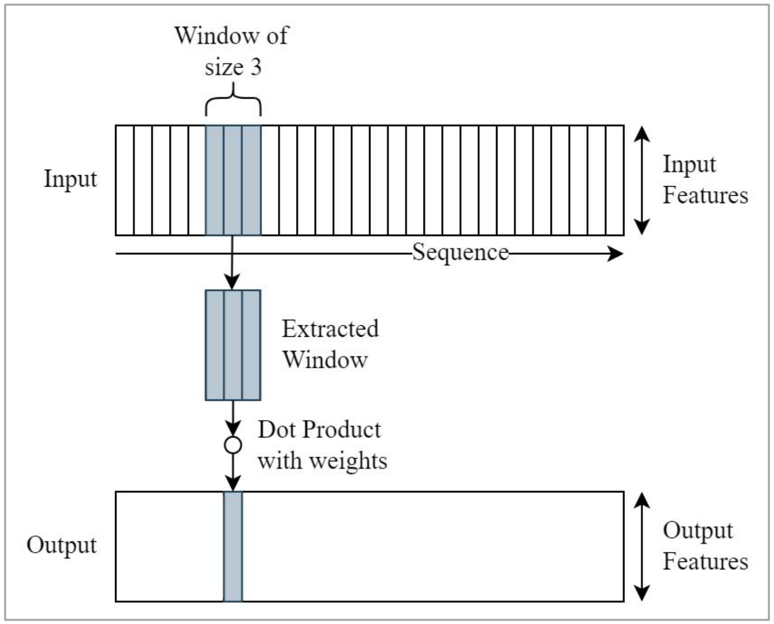


Figure 26: 1D Convolutional Neural Network

Summary of CNN Model

No	Layer Type	Kernel Size	Strides	Number of Filters/Units	Activation Function
1	Conv1D	3	1	64	relu
2	Batch Normalization	-	-	-	-
3	MaxPooling1D	2	2	-	-
4	Conv1D	3	1	128	relu
5	Batch Normalization	-	-	-	-
6	MaxPooling1D	2	2	-	-
7	Flatten	-	-	-	-
8	Dense	-	-	100	relu
9	Dropout	-	-	-	0.5
10	Dense	-	-	4	softmax

Table 1: Summary of CNN model

Deep Learning Approach: LSTM

What is LSTM?

- LSTM (Long Short-Term Memory)
- A type of recurrent neural network (RNN)
- RNNs tend to lose information over time
- LSTM can remember information for long periods
- Designed to classify sequential data
- Particularly suited for time series data

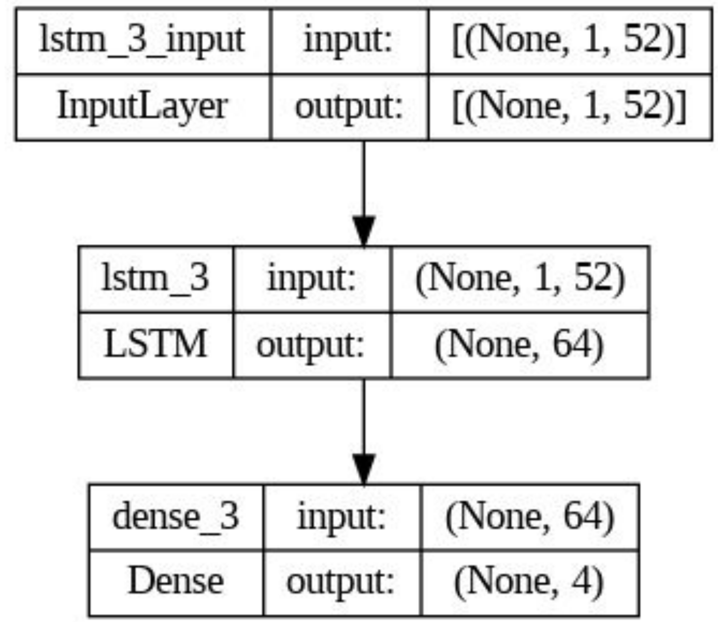
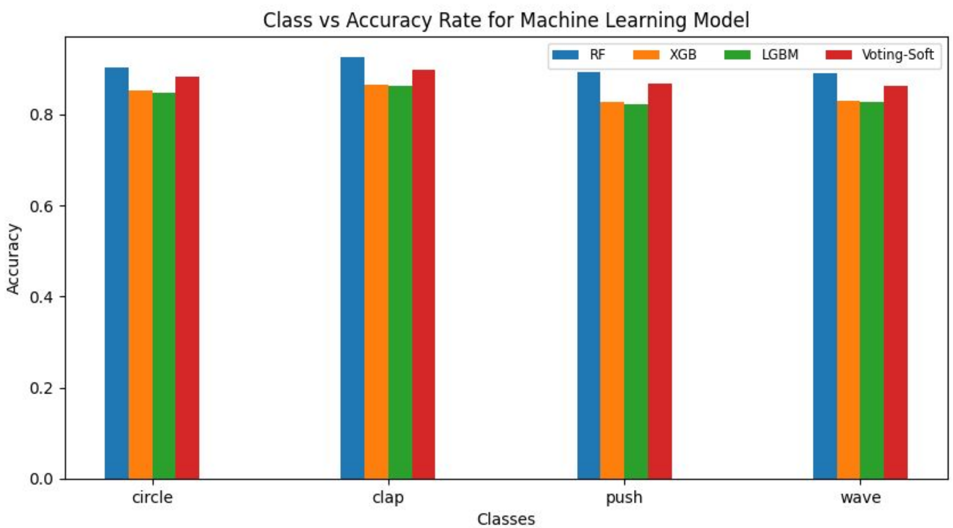


Figure 27: LSTM model architecture

Experimental Evaluation: Ensemble Learning Methods

Table 2: Performance of different Ensemble models

Model	Accuracy	Precision	Recall	F1_Score
Light-GBM	67.9%	68.0%	67.9%	67.8%
XGBoost	68.7%	68.9%	68.7%	68.6%
Random Forest	80.5%	80.6%	80.5%	80.4%
Voting Classifier	75.5%	75.7%	75.5%	75.4%



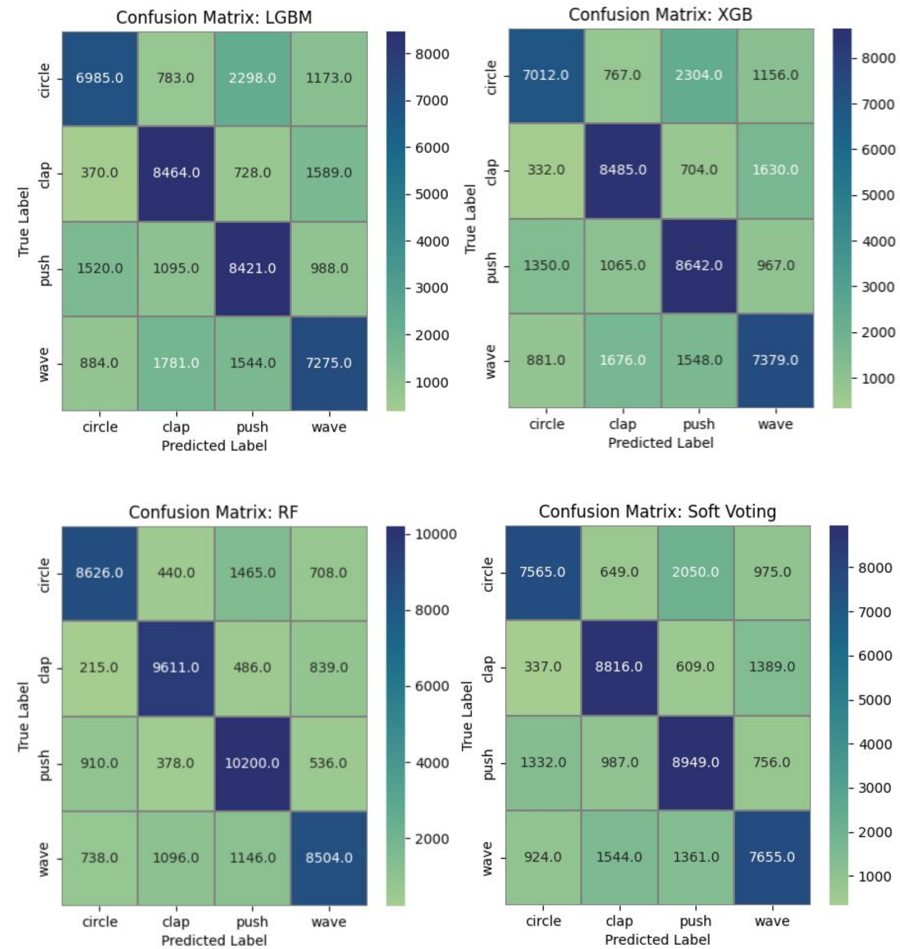


Figure 24: Performance Measures for Ensemble Models

Experimental Evaluation: Shapelet Learning Model

Table 3: Performance of Shapelet Learning model

Model	Accuracy	Precision	Recall	F1_Score
Learning Shapelet	74.5%	77.7%	74.5%	74.0%

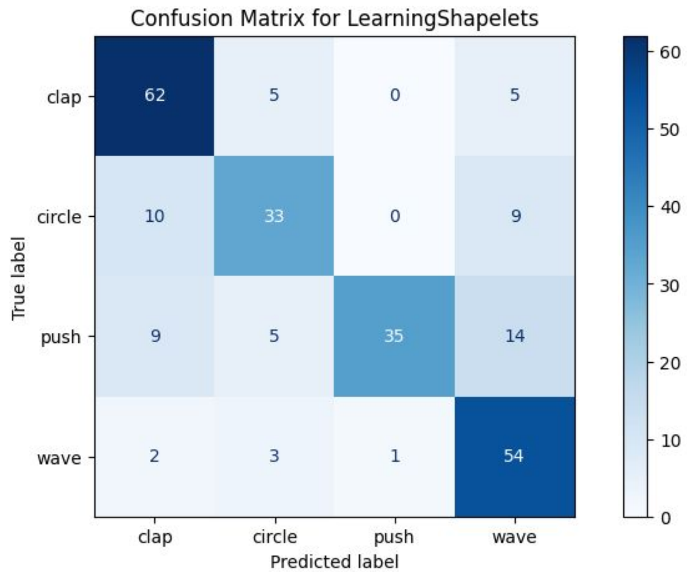


Figure 22: Performance of Shapelet Learning Model

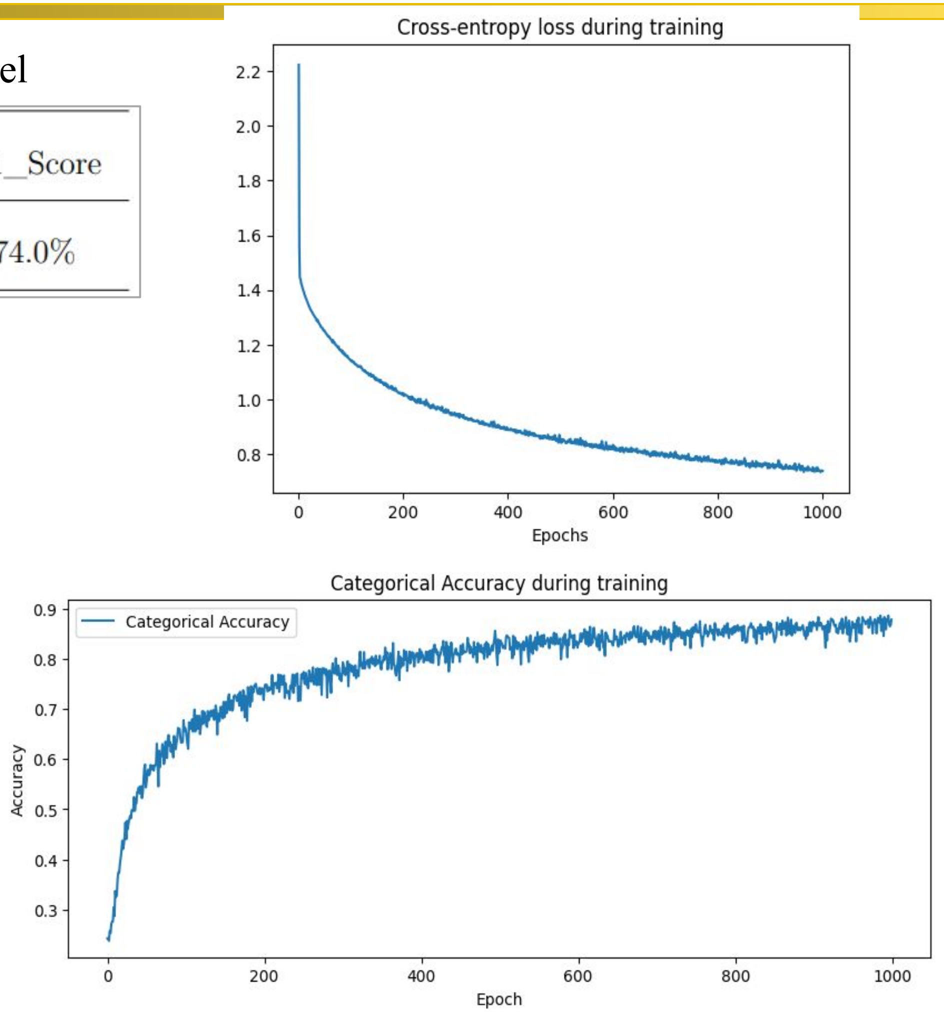


Figure 23: Evaluation of Loss & Accuracy during Training

Experimental Evaluation: CNN Model

Table 3: Performance of CNN model

Feature	Accuracy	Precision	Recall	F1_Score
Amplitude	73.31%	73.71%	73.31%	73.51%
Phase	59.77%	62.84%	59.77%	61.27%
Amplitude & Phase	76.55%	77.48%	76.55%	77.01%
Weighted-Average	78.13%	78.69%	78.13%	78.41%

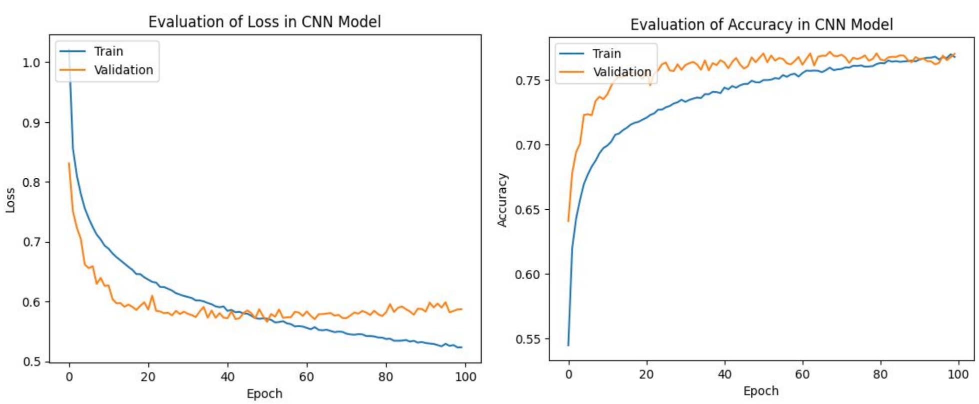


Figure 25: Evaluation of Loss & Accuracy during Training CNN model

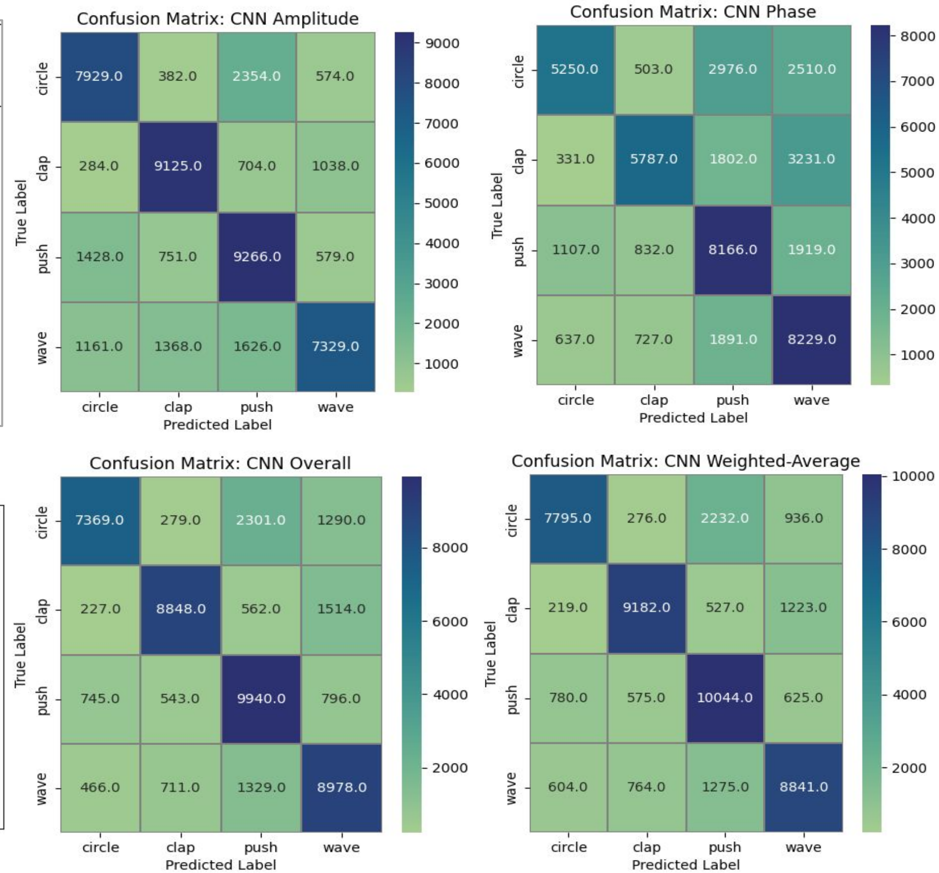


Figure 26: Performance of CNN Model with different features

Experimental Evaluation: LSTM Model

Table 4: Performance of LSTM model

Class	Accuracy	Precision	Recall	F1_Score
LSTM	81.11%	81.10%	81.11%	81.11%

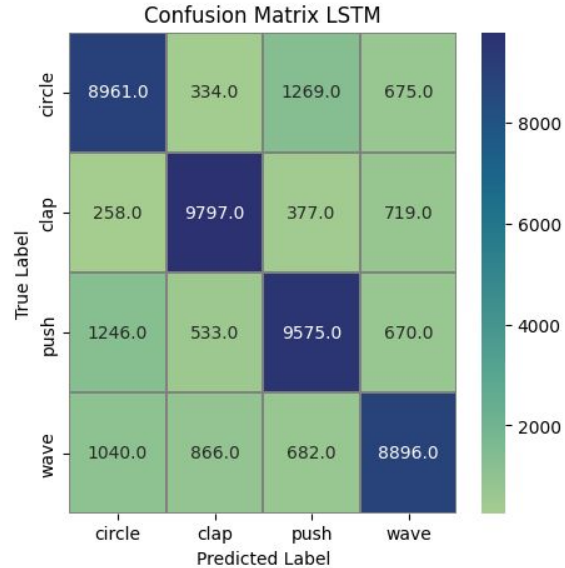


Figure 28: Performance of LSTM model

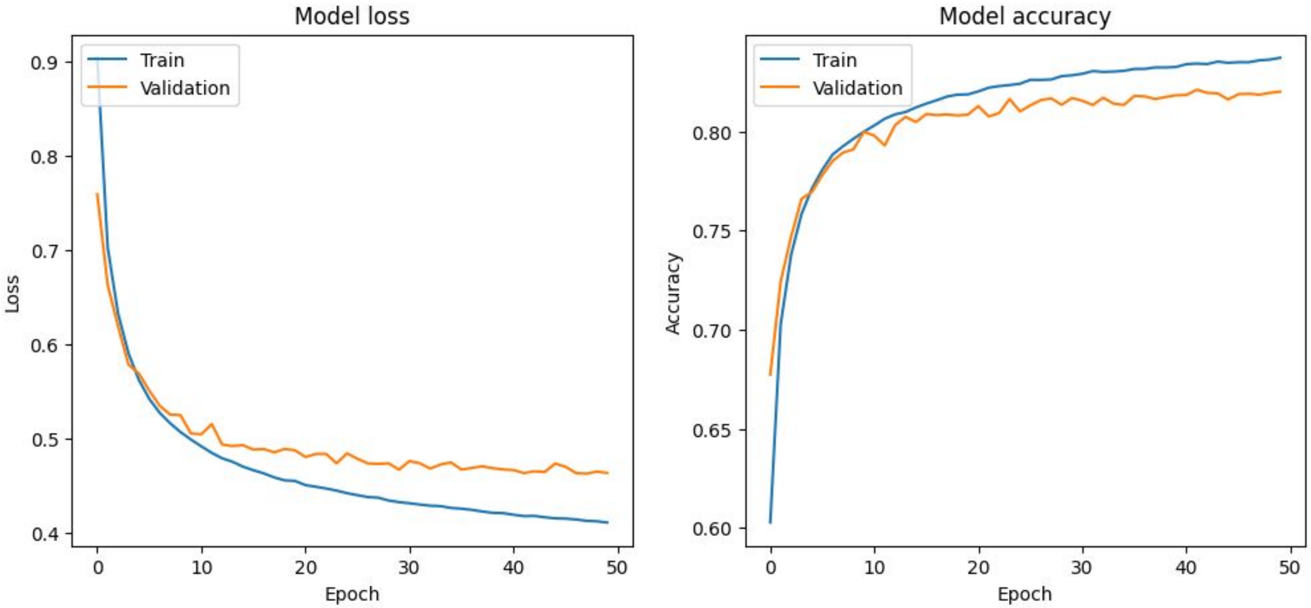


Figure 27: Evaluation of Loss and Accuracy during training

Comparison Analysis: Comparison among ML models

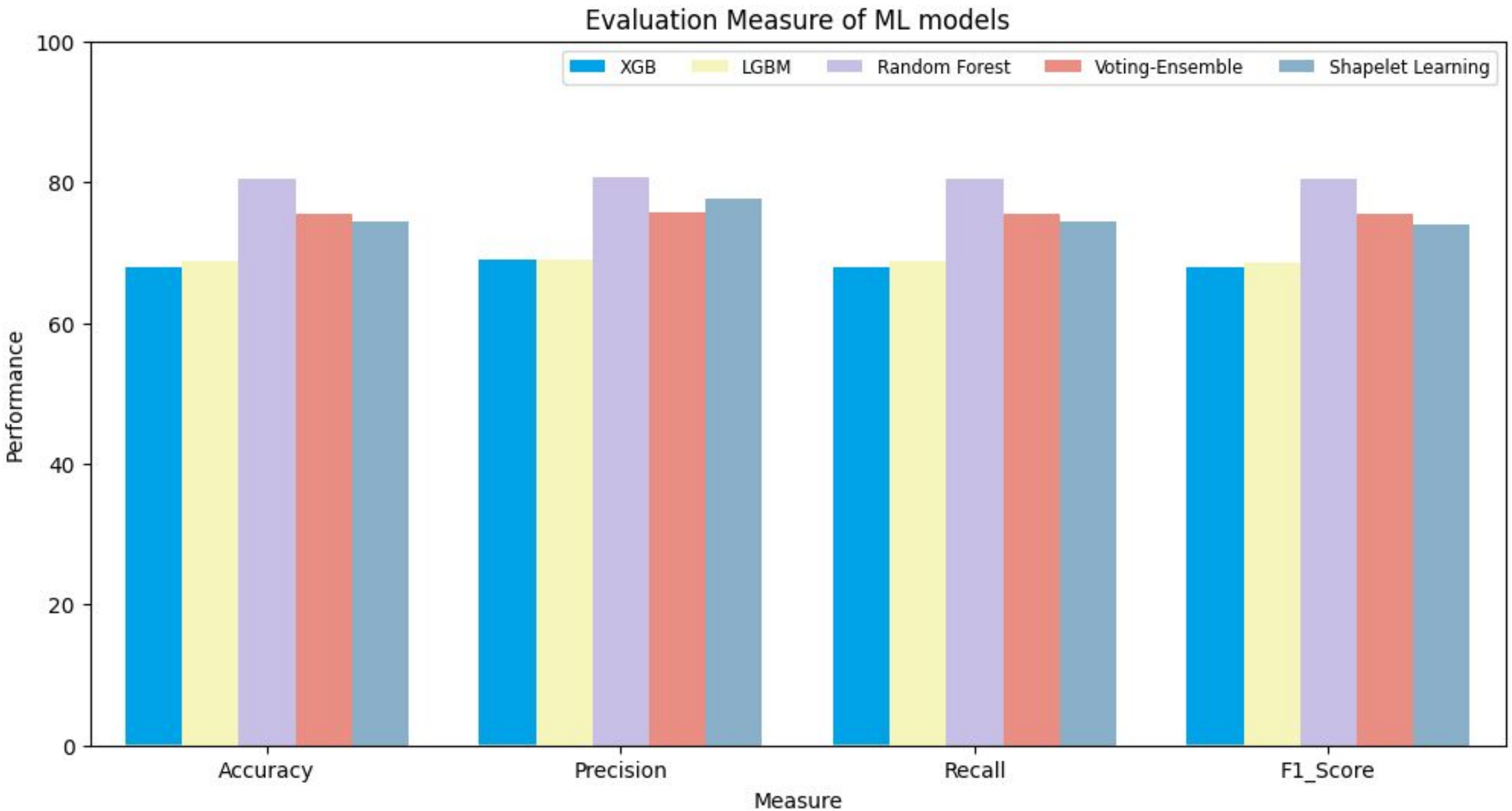


Figure 29: Comparison among ML models

Comparison Analysis: Comparison among DL models

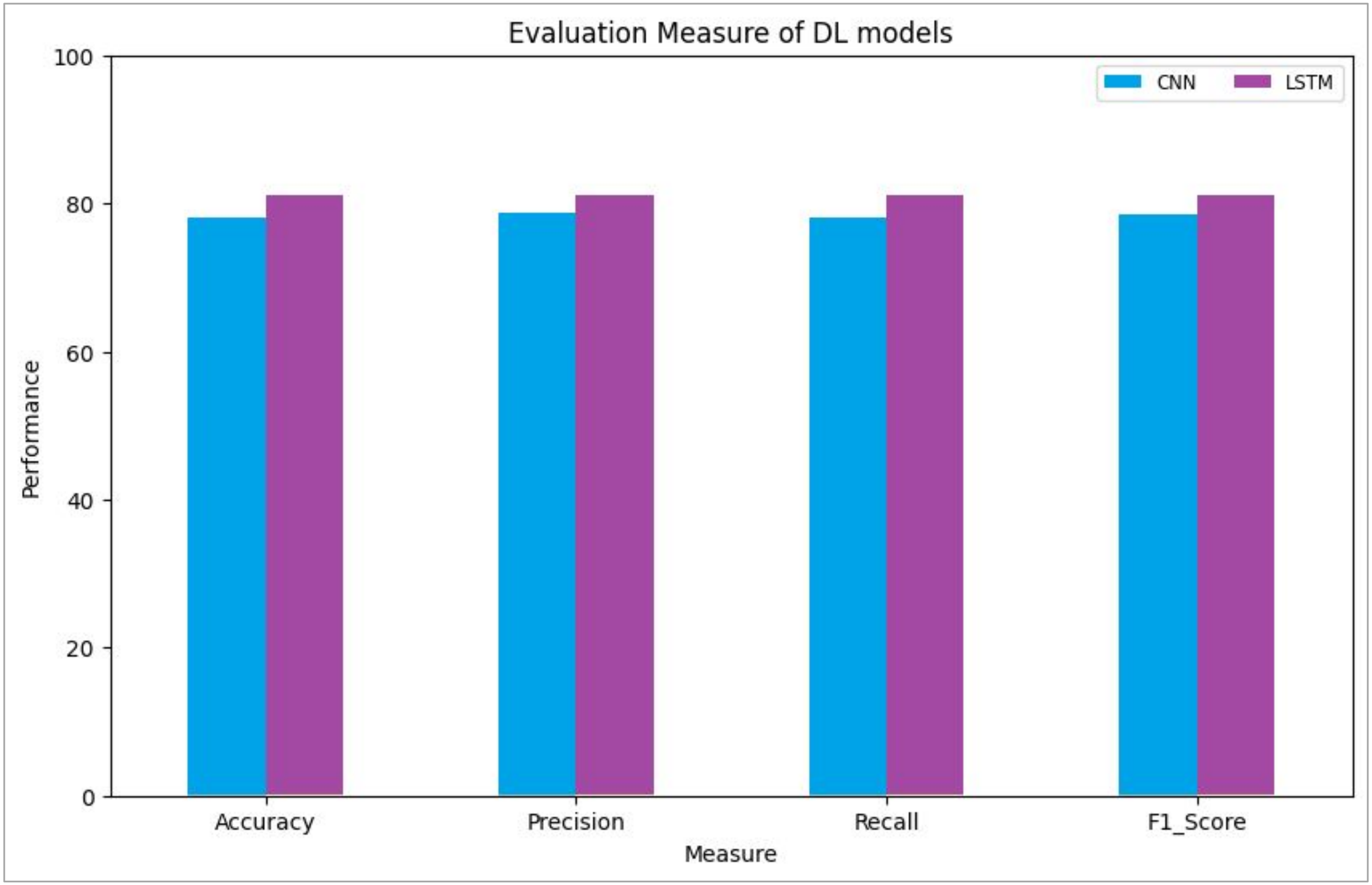


Figure 30: Comparison among DL models

Comparison Analysis: Comparison between best ML & DL model

- ❑ Random Forest(80.5% accuracy)
- ❑ LSTM(81.11% accuracy)
- ❑ LSTM surpasses Random Forest with slightly better performance.

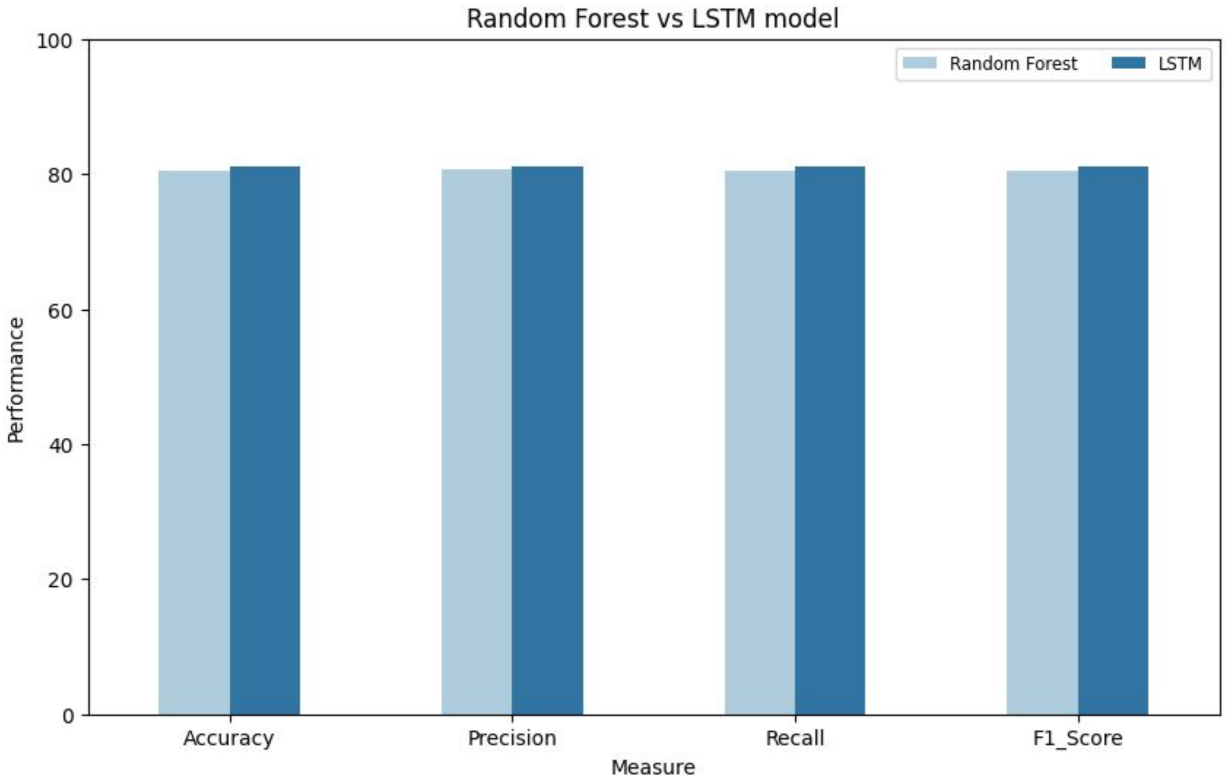


Figure 31: Comparison between Random forest and LSTM

Comparison Analysis: Comparison Between Datasets

- Dataset with Fixed point data only predicts more accurately for all models.

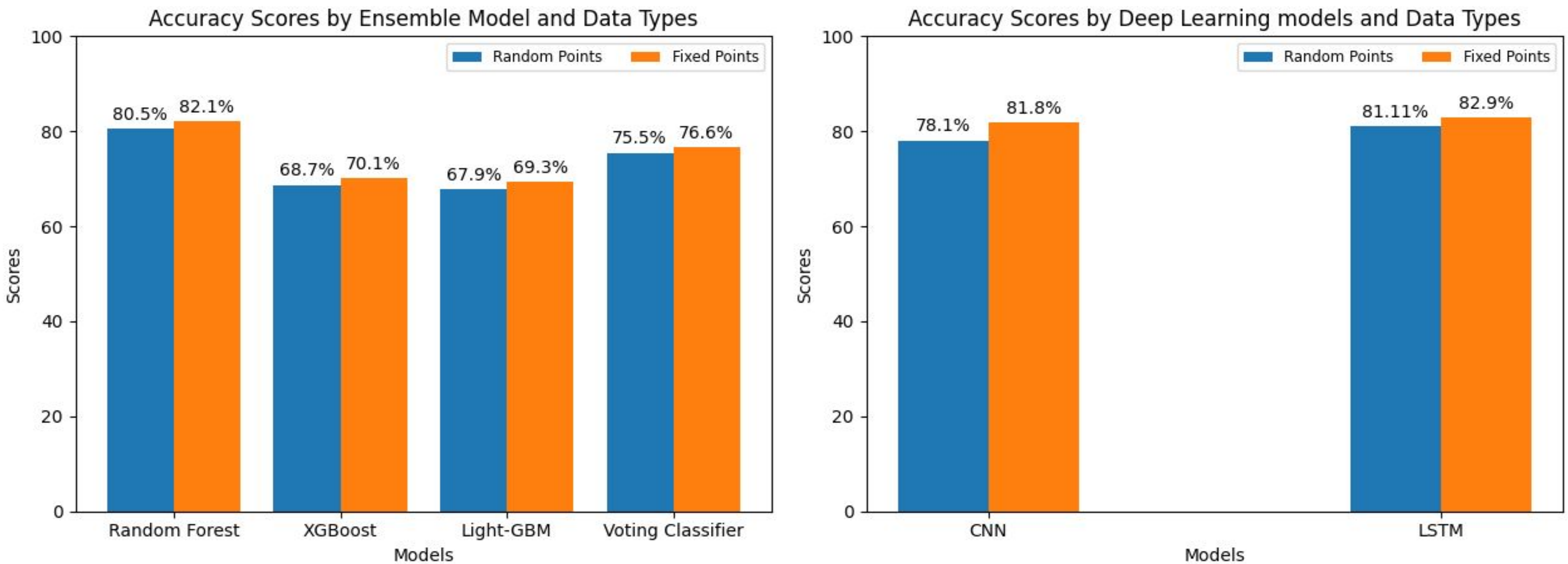


Figure 32: Comparison between models for dataset with Random points and Fixed points only

Comparison Analysis: Comparison With Existing Work

Table 4: Existing Methods vs Proposed Method

Author	Year	Tool	Method	Accuracy
Ohara et al.[3]	2018	Intel 5300 NIC	HMM	67.8%
Hasmath et al.[2]	2021	Intel 5300 NIC	LSTM	78.0%
Yong et al.[4]	2022	802.11 based CSI tool	Meta-learning	79.5%
Marwa et al.[1]	2023	ESP32	Random Forest	72%
Proposed Work	-	ESP32	LSTM	81.11%

Comparison Analysis: Comparison With Existing Work

- Our proposed model outperforms some other existing methods

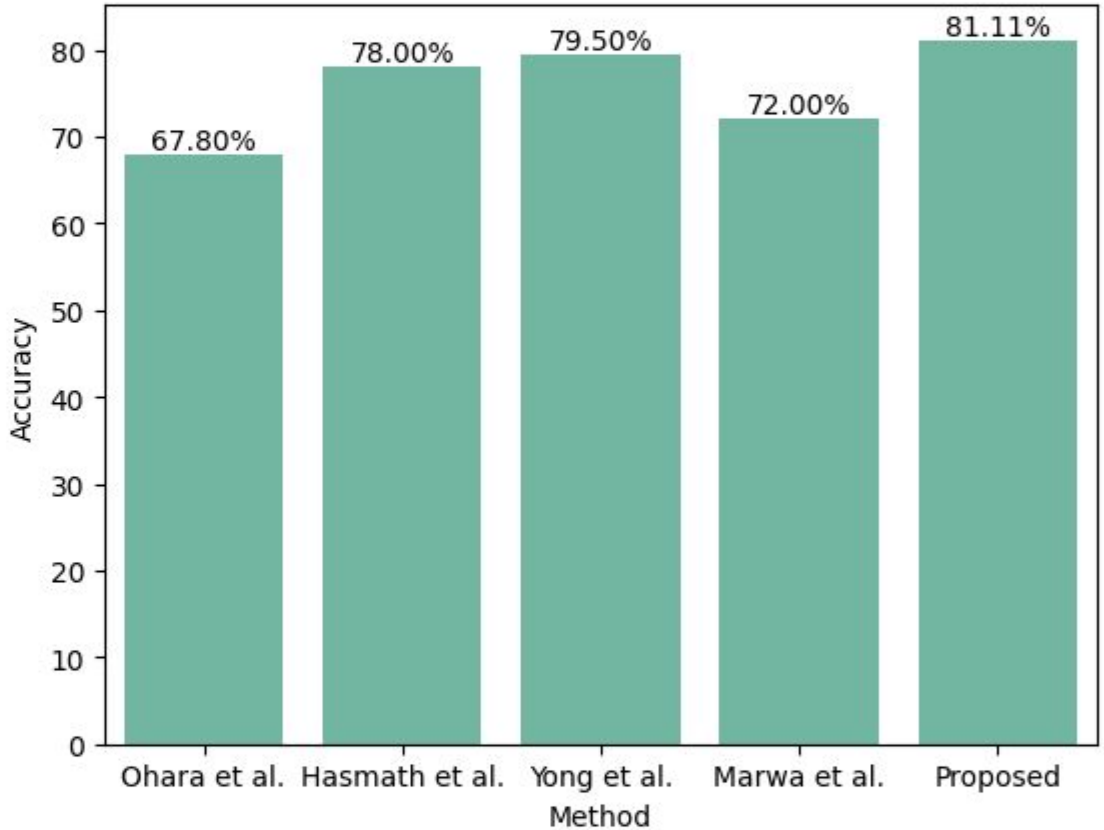


Figure 34: Comparison among existing vs proposed methods

Conclusion

- **CSI-based position-independent gesture recognition system**
- Best performance with LSTM model, using CSI amplitude data
- Applicable in operating smart home appliances at a low cost.

Limitations

- The system is trained & tested in a real-world environment, which might include some noise or disturbances.
- The system isn't yet designed to work effectively in rooms with many people, which could affect its accuracy.

Future Work

- Using more ESP32 microcontrollers for data collection.
- Gathering data with more people around to make the system more practical.
- Focusing more on the phase data and combined datasets.
- Applying advanced deep learning methods.

References

[1] Marwa R. M. Bastwesy, H. Choi and Y. Arakawa, ‘Tracking on-desk gestures based on wi-fi csi on low-cost microcontroller,’ in 2023 Fourteenth International Conference on Mobile Computing and Ubiquitous Network (ICMU), 2023, pp. 1–6. doi: 10.23919/ICMU58504.2023.10412222 (cit. on pp. 7, 37).

[2] Hasmath F. Thariq Ahmed, H. Ahmad, S. K. Phang, H. Harkat and K. Narasingamurthi, ‘Wi-fi csi based human sign language recognition using lstm network,’ in 2021 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), 2021, pp. 51–57. doi: 10.1109/ IAICT52856.2021.9532548 (cit. on pp. 8, 37).

[3] K. Ohara, T. Maekawa, S. Sigg and M. Youssef, ‘Preliminary investigation of position independent gesture recognition using wi-fi csi,’ in 2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), IEEE, 2018, pp. 480–483 (cit. on pp. 9, 37). 10.1109/INFOCOM.2014.6847958.

References(cont.)

[4] Y. Zhang, X. Wang, Y. Wang and H. Chen, ‘Human activity recognition across scenes and categories based on csi,’ IEEE Transactions on Mobile REFERENCES 84 Computing, vol. 21, no. 7, pp. 2411–2420, 2022. doi: 10.1109/TMC.2020. 3041756 (cit. on pp. 37).

[5] J. Grabocka, N. Schilling, M. Wistuba and L. Schmidt-Thieme, ‘Learning time-series shapelets,’ in Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, 2014, pp. 392– 401 (cit. on pp. 26).

THANK YOU!

Any Questions?