A Dissertation Report

On

Human Action Recognition Using Mobile Sensors

Submitted in partial fulfillment of the
requirements for the award of course credits for
Master of Technology

Submitted By:

Anmol Maheshwari

M.Tech CSE

Enrolment No. 17535004

Under the guidance of

Dr. Balasubramanian Raman

Professor

Department of Computer Science and Engineering

INDIAN INSTITUTE OF TECHNOLOGY, ROORKEE

Roorkee – 247667

May, 2019

DECLARATION

I declare that the work presented in this dissertation with title " Human Action Recognition Using Mobile Sensors" towards fulfillment of the requirement for the award of the degree of Master of Technology in Computer Science & Engineering submitted in the Department of Computer Science & Engineering, Indian Institute of Technology Roorkee, India is an authentic record of my own work carried out during the period of May 2018 to May 2019 under the supervision of Dr. Balasubramanian Raman, Professor, Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, Roorkee, India. The content of this dissertation has not been submitted by me for the award of any other degree of this or any other institute.

Date: ....

Place: Roorkee

ANMOL MAHESHWARI

(17535004)

M.TECH (CSE)

CERTIFICATE

This is to certify that the statement made by the candidate is correct to the best of my Knowledge and belief.

Date: ....

Place: Roorkee

Sign: ....

Dr. Balasubramanian Raman

(Professor)

Indian Institute of Technology

Roorkee-247667

ABSTRACT

Mobile phones now days have different kinds of sensors such as accelerometer, gyroscope, proximity sensors, barometer etc. Nowadays most people have smart phones and almost all smart phones have these sensors. So data collected with these sensors can help to do some interesting thing like human action recognition. Now our phones has tri-axial accelerometer (X, Y and Z axis) and a tri-axial (X, Y and Z axis) gyroscope that measures linear acceleration and angular velocity on all three axis respectively. Data collected with these sensors can help in human activity recognition, in real time, and is quite a significant challenge for those attempting to find out calories burnt, tracking hours slept and so forth. Smart watches also have gyroscope and accelerometer to track human activity, calories burnt etc. These watches have other sensors like heart rate monitor to do or provide additional functionalities. This report compares the result of various classical machine learning paradigm such as SVM, Logistic Regression etc on engineered features and LSTM, GRU etc on raw data. Along with comparison, the ensemble model is also designed using best models among LSTM, GRU, SVM, Logistic Regression etc that aim to produce more better and improved results than the existing models and to be the state-of-art.

ACKNOWLEDGMENT

Dedicated to my family and friends, for standing by me through thick and thin, without whom i would not have gotten this far. I would like to express my sincere gratitude to my advisor Dr. Balasubramanian Raman for the continuous support of my study and research, for his patience, motivation, enthusiasm and immense knowledge. His guidance helped me in all time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my study.

I am also grateful to the Department of Computer Science and Engineering and Research Scholars Lab, IIT Roorkee for providing valuable resources to aid my research.

ANMOL MAHESHWARI

TABLE OF CONTENTS

DECLARATION **ii**

CERTIFICATE **iii**

ABSTRACT **iv**

ACKNOWLEDGEMENT **v**

1 INTRODUCTION **1**
1.1 *Research Gap and Our Contribution* 1

2 LITERATURE SURVEY **3**
2.1 *Human Activity Recognition Using Accelerometer and Gyroscope Sensors* 3
2.2 *Human Activity Recognition Using Smartphone Sensors with Context Filtering* 3

3 PROBLEM FORMULATION **4**
3.1 *Problem Definition* 4
3.2 *Preliminaries* 4
3.2.1 Univariate Analysis 4
3.2.2 t-SNE 4
3.2.3 Logistic Regression 5
3.2.4 Linear Support Vector Classifier 5
3.2.5 Kernel Support Vector Classifier 5
3.2.6 Long Short Term Memory (LSTM) 5
3.2.7 Gated Recurrent Unit (GRU) 6
3.2.8 Bi-Directional Recurrent Neural Networks (BRNN) 7
3.2.9 Ensemble Learning 7

4 PROPOSED APPROACH **9**
5 EXPERIMENTATION AND RESULTS **11**
5.1 *Data-set* 11
5.2 *Evaluation Metrics Used* 12

5.3 *Results* 14
5.3.1 LSTM 14
5.3.2 GRU 15
5.3.3 Bi-Directional LSTM 16
5.3.4 Bi-Directional GRU 17
5.3.5 Logistic Regression 18
5.3.6 Linear SVC 19
5.3.7 RBF SVC 20
5.3.8 Ensemble Techniques 21
5.3.8.1 ***Ensemble Method with MAX Voting*** 22
5.3.8.2 ***Ensemble Method with Sum Rule*** 23
5.3.8.3 ***Ensemble Method with Product Rule*** 24
5.3.8.4 ***Ensemble Method with Mean Rule*** 25

6 CONCLUSIONS AND FUTURE SCOPE 27

REFERENCES 28

LIST OF TABLES

Table 5.1	Comparison of Accuracy of deep learning models	18
Table 5.2	Comparison of Accuracy of classical machine learning models	21
Table 5.3	Comparison of All Techniques Implemented	26

LIST OF FIGURES

Figure 3.1	t-SNE block diagram	4
Figure 3.2	LSTM	6
Figure 3.3	GRU	6
Figure 3.4	Bidirectional RNN	7
Figure 4.1	Basic LSTM Architecture for multi class classification based on time series data	9
Figure 4.2	Overall Architecture	10
Figure 5.1	Converting Total Acceleration to Body Acceleration on Respective Axis	11
Figure 5.2	Converting time series data to 128 size Vector for Respective Feature	12
Figure 5.3	t-SNE Analysis with perplexity=50	12
Figure 5.4	Univariate Anaylsis on tBodyAccMagmean	12
Figure 5.5	Confusion Matrix	13
Figure 5.6	Confusion Matrix for LSTM	14
Figure 5.7	Classification Report for LSTM	14
Figure 5.8	Confusion Matrix for GRU	15
Figure 5.9	Classification Report for GRU	15
Figure 5.10	Confusion Matrix for Bidirectional LSTM	16
Figure 5.11	Classification Report for Bidirectional LSTM	16
Figure 5.12	Confusion Matrix for Bidirectional GRU	17
Figure 5.13	Classification Report for Bidirectional GRU	17
Figure 5.14	Confusion Matrix for Logistic Regression	18

Figure 5.15	Classification Report for Logistic Regression	19
Figure 5.16	Confusion Matrix for Linear SVC	19
Figure 5.17	Classification Report for Linear SVC	20
Figure 5.18	Confusion Matrix for RBF SVC	20
Figure 5.19	Classification Report for RBF SVC	21
Figure 5.20	Confusion Matrix for MAX Vote Ensemble Technique	22
Figure 5.21	Classification Report for MAX Votes Ensemble Technique	22
Figure 5.22	Confusion Matrix for Sum Rule Ensemble Technique	23
Figure 5.23	Classification Report for Sum Rule Ensemble Technique	23
Figure 5.24	Confusion Matrix for Product Rule Ensemble Technique	24
Figure 5.25	Classification Report for Product Rule Ensemble Technique	24
Figure 5.26	Confusion Matrix for Mean Rule Ensemble Technique	25
Figure 5.27	Classification Report for Mean Rule Ensemble Technique	25

CHAPTER 1 INTRODUCTION

Long short-term memory (LSTM) is kind of artificial recurrent neural network (ARNN), [3] that is used in the field of deep learning to learn and forecast long and short sequences. LSTM have both feed-forward and feedback connections that make it act like a "general purpose computer". Given required time LSTM can do any computation that a Turing machine can perform [8]. A LSTM unit consists of a cell, an input gate, an output gate and a forget gate. The cell can remember and learn values over certain time intervals and information flow inwards and outwards of the cell is regulated by three gates. LSTM networks using time series data are befitted for processing, classifying and making predictions, since as we are aware of the fact that even important events can have lags of little duration in the data. LSTMs were basically refined to adapt the exploding and vanishing gradient problems that are faced during while training out traditional RNNs [21]. Gated recurrent units (GRUs) as the name itself suggests are type of gating mechanism in recurrent neural networks, introduced by Kyunghyun Cho et al in 2014.[4] The GRU is like a long short-term memory unit with a forget gate[5] in place of output gate, so it has few trainable parameters than LSTM[18]. GRU's in terms of performance works similar to LSTM in jobs like speech signal modeling and polyphonic music modeling. On certain smaller datasets, GRUs has displayed even better performance.[17] GRU is obtained by combining three gates of LSTM into two gates: update gate (z_t) and reset gate (r_t). Bidirectional Recurrent Neural Networks (BRNN) is used to connect two or more hidden layers of different directions towards the same output. With this form of generative deep learning, both the past and future states is available to the output layer at same time. Invented by Paliwal and Schuster[13] in 1997, the amount of data available to neural network is extended by BRNNs. BRNNs used the idea of duplicating the first recurrent layer, then provide the input sequence as input to first layer and input sequence’s reversed copy to the following layer. Ensemble methods provides better predictive performance as it combines multiple learning algorithms rather than using a single classifier [14][19][20]. It gives one optimal predictive model by combining all the good features of the constituent classifiers.

1.1 Research Gap and Our Contribution

We found a wide gap between results obtained in existing papers and the results of state of art approaches. The research paper [1] suggests that while dynamic activities can be efficiently classified, but the non-dynamic actions have various overlaps. Like the standing activity is overlapped with the sitting activity. So it requires further study of inputs that are available and revision of process pipeline phases of HAR. And other gap was that they used engineered features that were generated with the help of the domain expert. Now since we have deep learning models, we don’t need any feature engineering, our model will learn itself. Thus we built some deep learning models that would just use the raw data to directly to classify the various activities like Walking, Walking_Upstairs, Walking_Downstairs, Sitting, Standing and Laying. Our purpose was to get comparable results with the baseline models without any feature engineering being involved. And then after comparison of classical machine paradigm with deep learning paradigm, an ensemble of both deep learning model and classical machine learning model is designed to overcome cons of both methods that even give better results than baseline models and to be the state-of-art.

CHAPTER 2 LITERATURE SURVEY

2.1 A Public Domain Dataset for Human Activity Recognition Using Smartphones [1].

Computing centered on human is a rising research field that is based on understandings of human behavior. It integrates computer systems with users along with their social context. Sensing human body actions using smartphones to get context information about people's actions has turned out to be one of the appealing and challenging applications in this framework.. In this context, author described the work with an database of Activity Recognition, built from 30 subjects while carrying a waist-tied smart-phone with set in inertial sensors and is released for public interest on a well-known repository. Using the data collected from sensors like accelerometer and gyroscope they engineered 561 features and then they applied classical machine learning algorithm SVM and got the accuracy of around 96%.

2.2 Human Activity Recognition using Smartphone Sensors with Context Filtering [9].

Nowadays application of Ambient Intelligence e.g. assisted healthcare, remote monitoring and smart home, with the help of smart phones to record human activities has become a topic of high interest. Simple activities like walking, running, sitting can be perceived easily but more-complex activities like moving up and down the stairs, jogging, slow running, fast running are often difficult to recognize accurately. Author aimed to reduce the error rate of recognizing these kinds of activities by introducing context filtering and applying Dynamic Time Warping (DTW) algorithm. They used atmospheric pressure sensor data and heart rate data as part of context filtering. They used a steady state of object as initial template and used this steady state with every activity. On the score of DTW, K Nearest Neighbor classification model was applied to get optimal value of threshold. Use of context filtering approach was made to further distinguish activities after primary classification of activities and thus removing confusions. In their study, they have observed that accuracy has significantly increased for discrepating similar kinds of activities. Overall, their approach has shown considerable performance improvements by applying context filtering and DTW algorithm shows at a lower cost.

CHAPTER 3 Problem Formulations

3.1 *Problem Definition*

Now a day’s mobile phones incorporates various sensors such as accelerometer, gyroscope, proximity sensors, barometer etc. Nowadays most people have smart phones and almost all smart phones have these sensors. So data collected from mobile’s sensor can help to do some interesting thing like human action recognition. Now our problem boils down to given raw time series data as accelerometer and gyroscope data from our phone, we need to build a model, to train over data set and give comparable result as we were using engineered features [1] to classify the activity into one of the mentioned six classes that is walking, standing, laying, walking downstairs, walking upstairs and sitting. Finally we have to obtain a model that works better than our baseline model [1].

3.2 *Preliminaries*

3.2.1 Univariate Analysis

It means just one variable or one feature analysis. In machine learning, univariate analysis means to construct histogram using just one variable that is one feature and see the difference between various classes on the basis of that one particular feature. Sometimes just maybe one feature may be powerful enough to separate different classes.

3.2.2 t-SNE

T-Distributed Stochastic Neighborhood Embedding is a technique used for dimensionality reduction. It one of the best dimensionality reduction paradigm. It is used for visualization of dataset. Basically it converts d-dimension(High) data set to low dimension dataset [6].

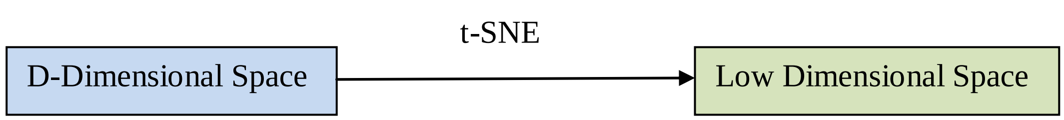


Figure 3.1 t-SNE block diagram

3.2.3 Logistic Regression

The logistic regression was developed primarily by Joseph Berkson, [10] beginning in Berkson (1944), where he invented "logit". In regression analysis, logistic regression is estimating the variables of a logistic model. It is a classification technique which is used to classify different points in data to respective classes. It is a form of binomial regression which is represented by an indicator variable, which can have only two possible values "1" and "0".

3.2.4 Linear Support Vector Classifier(SVC)

In machine learning, SVC are supervised learning models with associated learning algorithms that analyze data for classification. The main motive of SVC is to increase gap between separate categories points after mapping them into space. On whichever part of the gap a new examples falls decides their predicted category [11].

3.2.5 Kernel Support Vector Classifier

The kernel function in SVC means transforming data into another dimension in order to get a definite dividing margin between different data classes [15]. The kernel function is applied on all pair of data instance for the mapping of the original non-linear observations to a feature space of higher-dimensional in which they become separable. Finding out the coordinates of the data in that space is found to be computationally expensive than using the kernel function [16].

3.2.9 Long Term Short Memory (LSTM)

Long short-term memory (LSTM) is kind of artificial recurrent neural network (ARNN), [3] that is used in the field of deep learning to learn and forecast long and short sequences. LSTM have both feed-forward and feedback connections that make it act like a "general purpose computer". Given required time LSTM can do any computation that a Turing machine can perform [8]. A LSTM unit consists of a cell, an input gate, an output gate and a forget gate. The cell can remember and learn values over certain time intervals and information flow inwards and outwards of the cell is regulated by three gates. LSTM networks using time series data are befitted for processing, classifying and making predictions, since as we are aware of the fact that even important events can have lags of little duration in the data. LSTMs were basically refined to adapt the exploding and vanishing gradient problems that are faced during while training out traditional RNNs [21]. The equations for LSTM are stated below:-

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right), \tag{1}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i), \tag{2}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o), \tag{3}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma(W_c x_t + U_c h_{t-1} + b_c), \tag{4}$$

$$h_t = o_t \circ \sigma(c_t), \tag{5}$$

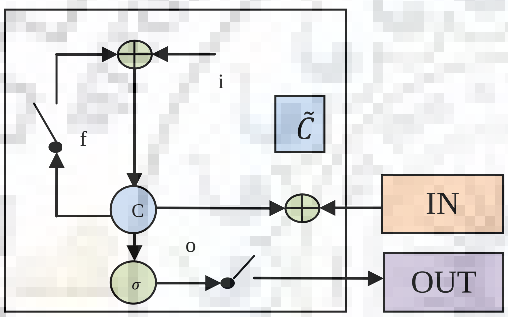


Figure 3.2 LSTM

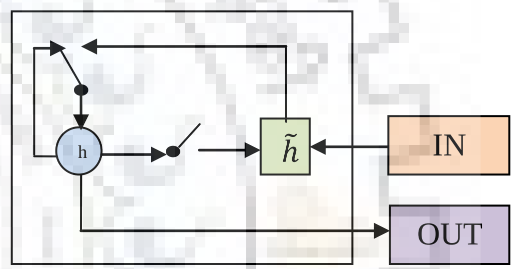


Figure 3.3 GRU

3.2.10 Gated Recurrent Unit (GRU)

Gated recurrent units (GRUs) as the name itself suggests are type of gating mechanism in recurrent neural networks, introduced by Kyunghyun Cho et al in 2014.[4] The GRU is like a long short-term memory unit with a forget gate[5] in place of output gate, so it has few trainable parameters than LSTM[18]. GRU's in terms of performance works similar to LSTM in jobs like speech signal modeling and polyphonic music modeling. On certain smaller datasets, GRUs has displayed even better performance [17]. As above in Figure 3.2b, GRU is obtained by combining three gates of LSTM into two gates: update gate (z_t) and reset gate (r_t). The equations of the GRU are given below:-

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z), \tag{6}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r), \tag{7}$$

$$h_t = (1 - z_t) o h_{t-1} + z_t o \sigma(W_h x_t + U_h (r_t o h_{t-1}) + b_h), \tag{8}$$

3.2.11 Bi-Directional Recurrent Neural Networks (BRNN)

Bidirectional Recurrent Neural Networks (BRNN) is used to connect two or more hidden layers of different directions towards the same output. With this form of generative deep learning, both the past and future states is available to the output layer at same time. Invented by Paliwal and Schuster in 1997 [13], the amount of data available to neural network is extended by BRNNs. BRNNs used the idea of duplicating the first recurrent layer, then provide the input sequence as input to first layer and input sequence's reversed copy to the following layer. These layers have LSTM or GRU unit.

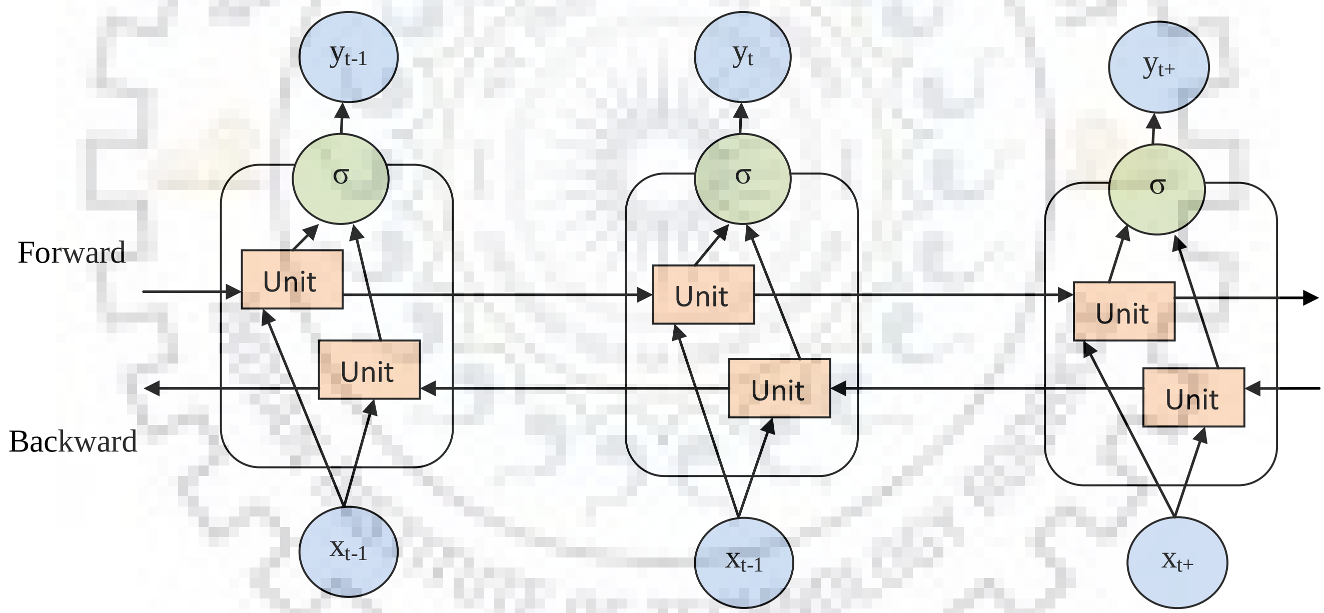


Figure 3.4 Bidirectional RNN

3.2.12 Ensemble Learning

Ensemble methods provides better predictive performance as it combines multiple learning algorithms rather than using a single classifier [14][19][20]. It gives one optimal predictive model by combining all the good features of the constituent classifiers. We have various ensemble techniques, some of them are Max Voting, Mean Rule, Sum Rule, Product Rule etc.

Mean Rule: $\mu_j(x) = \frac{1}{T}\sum_{t=1}^{T} d_{t,j}(x)$ (9)

Product Rule: $\mu_j(x) = \prod_{t=1}^{T} d_{t,j}(x)$ (10)

Sum Rule: $\mu_j(x) = \sum_{t=1}^{T} d_{t,j}(x)$ (11)

In above equations t=1,..,T where T is total count of classifiers and j=1,…,C where C is total count of classes.

CHAPTER 4 PROPOSED APPROACH

Given our data set with 7352 training data each of 128 size vector and 2947 test data, is given to some recurrent neural network that works on time series data. For that we have various RNN as LSTM, GRU, Bidirectional LSTM and Bidirectional GRU. Now the 128 sized vector is given as input to the model, they output one of the six class as output.

We have performed the experiments for raw data that was 9 time series data with some basic deep learning models. The basic model with single layer with of x unit of LSTM, GRU or Bidirectional LSTM or Bidirectional GRU layer is used. The input to the model is a 128 sized vector that is with 128 time steps. Now to hyper-tune the model GridSearchCV and Hyperas was used. The training data size is 7352 that is 7352 such windows. Then we fed the output from RNN to a dense layer which is basically a six class based classifier.

Now our problem of classification is done using the architecture below. The class labels are described below:-

1) Class 0- Walking
2) Class 1- Walking_Upstarirs
3) Class 2- Walking_Downstairs
4) Class 3- Sitting
5) Class 4- Standing
6) Class 5- Laying

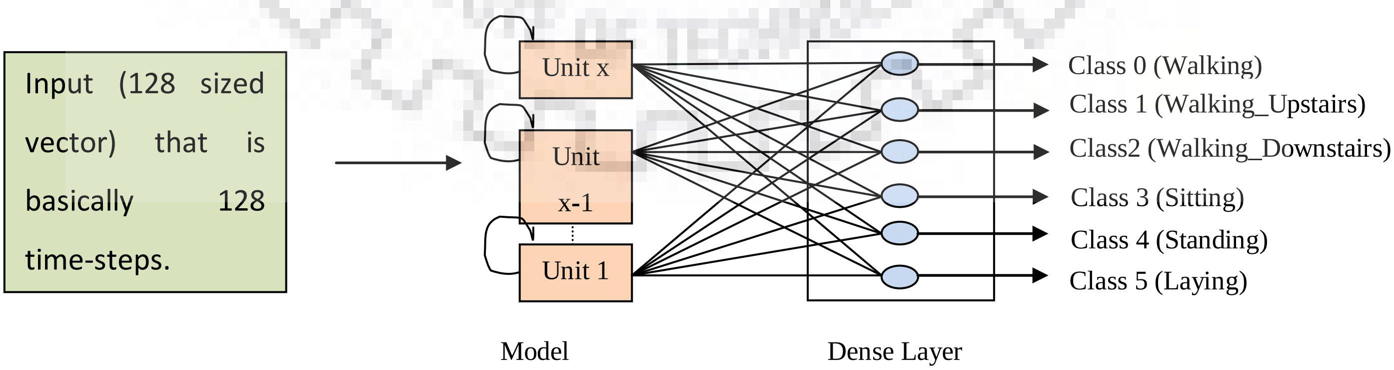


Figure 4.1 Basic RNN Architecture for multi class classification based on time series data.

This 6 class based classifier proposed works okay with no knowledge of the inputs fed to the model. That is the previous model, described in this paper [1] uses only the classical models on 561 engineered features designed with expert domain for the same.

But with our basic RNN model we don't need any engineered features, just the raw time series data as input. So it can directly work on the data and instant results with little preprocessing of our given input.

Now at first, hyper-tuning of our model is done to find out all the hyper parameters of our model that gives the best answer. The hyper-tuning is done using grid search and hyperas (wrapper for keras and hyperopt). Then the model is trained with the optimized parameters.

Now after the comparison of different classical models and deep learning models, ensemble learning is done, the models with high accuracy were chosen from both category. Then we did comparison of various ensemble techniques as max voting, sum rule, product rule and mean rule. And all the results were found out to better than any individual model.

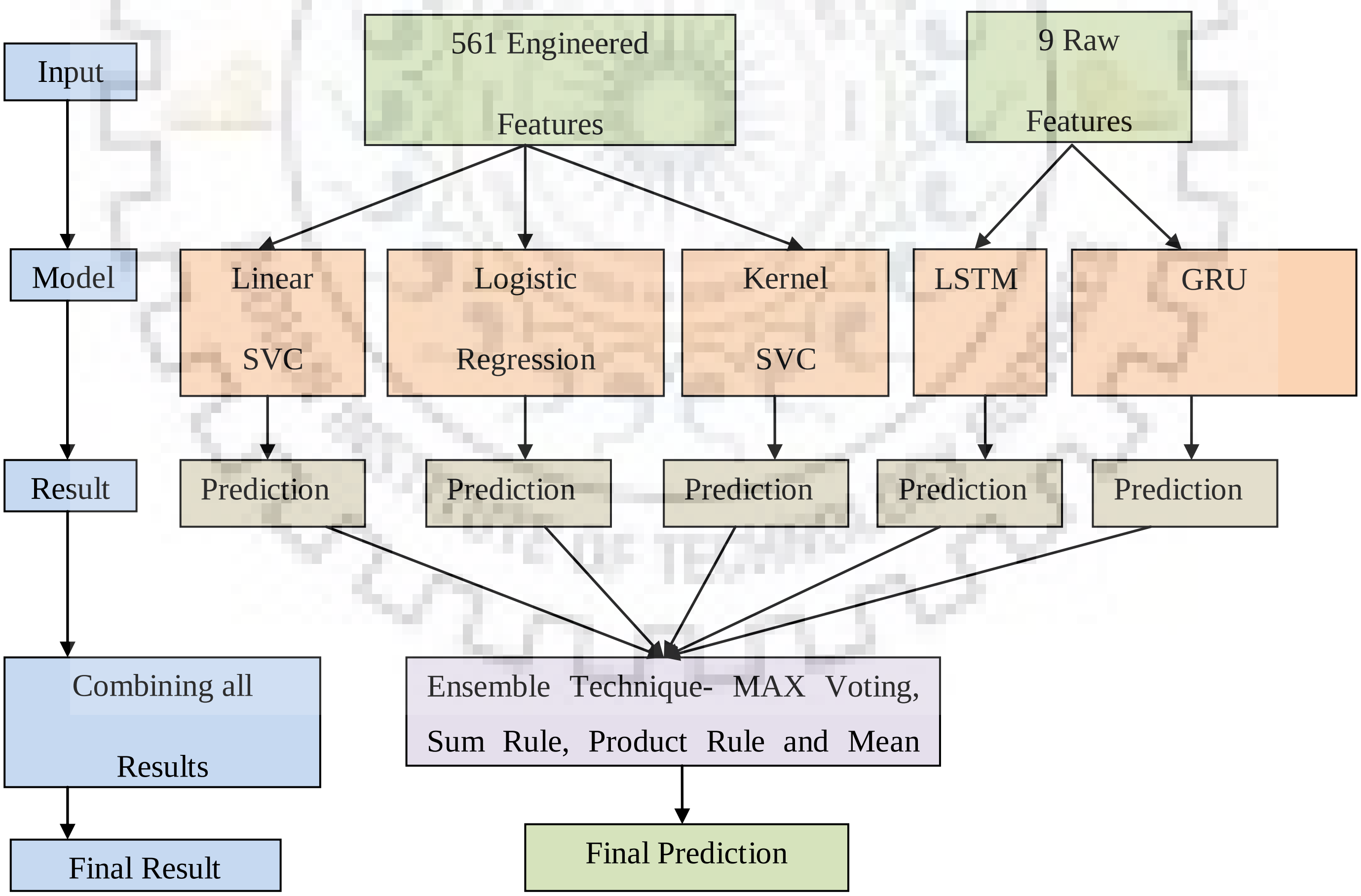


Figure 4.2 Overall Architecture.

CHAPTER 5 EXPERIMENTATION AND RESULTS

5.1 *Data-set*

Human Activity Recognition Using Smart-phones Data Set [1]. Given our data set with 7352 rows of training data and 2947 test data, each of 128 size vector. The data is recorded with Gyroscope and Accelerometer on all 3 axis that generates data as time signals, which is angular velocity and acceleration with respect to time. So we have total 6 signals that are gyroscope and accelerometer readings on all 3 axis, X, Y and Z. Now acceleration signal has both body motion and gravitational components and if we apply Butterworth loss [2] pass filter on it, it separates gravitational force and body acceleration as it is assumed that gravitational component have low frequency component. Now given total raw 9 time series data, the basic approach that could be used is any Recurrent Neural Network.

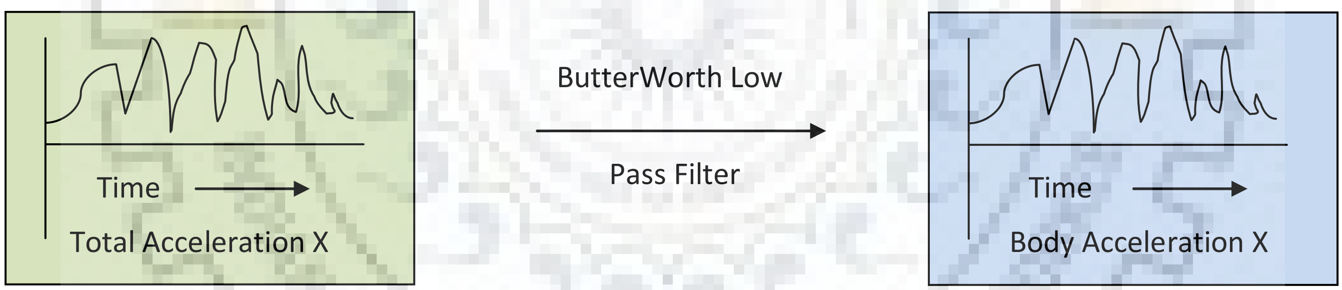


Figure 5.1 Converting Total Acceleration to Body Acceleration on Respective Axis.

Now as above filter is applied on X axis, it is also applied on other axis data as well. Now, below diagram clearly states how the given time signal data is converted to 128 size vector by using fix window of 1.28 seconds and the next window will start from 0.64 seconds and will overlap till 1.28 second with previous window and then will be present up to 1.92 seconds. And then the next window starts from 1.28 seconds and so on. That means there is total of 50% overlap of window with the previous window. The diagram for the same is given below:-

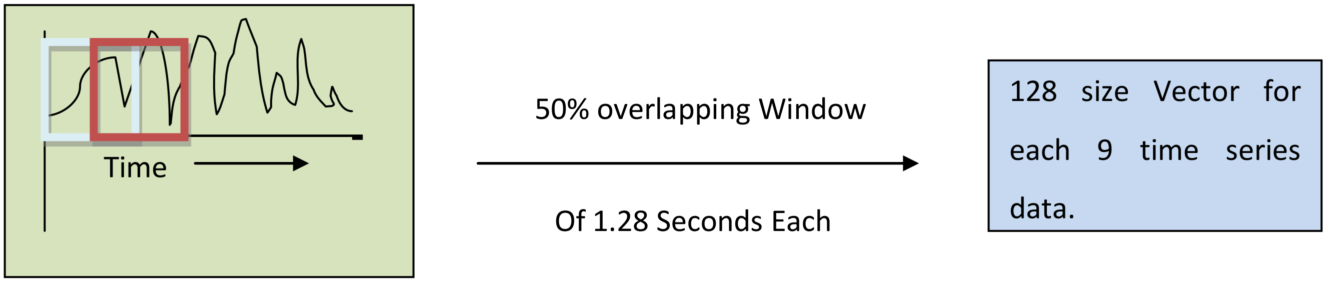
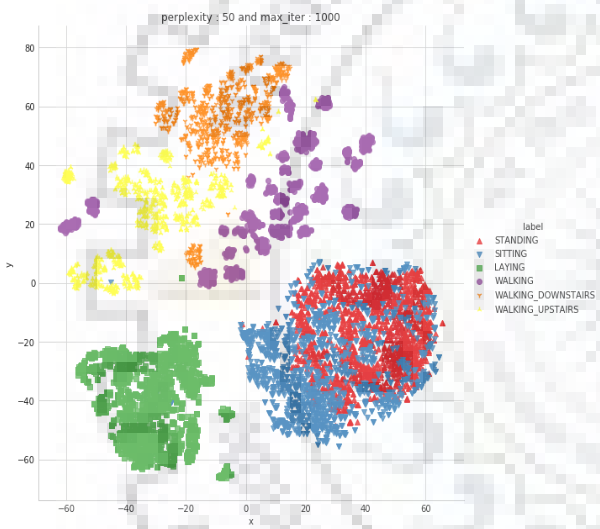
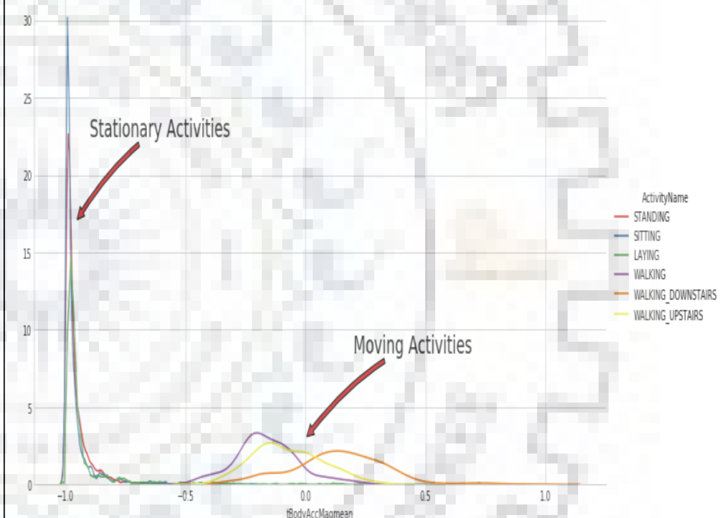


Figure 5.2 Converting time series data to 128 size Vector for Respective Feature.

T-SNE and Univariate Analysis using features that were constructed with feature engineering in [1].

T-SNE with perplexity=50	Univariate Analysis on tBodyAccMagmean
 Figure 5.3 t-SNE with perplexity=50	 Figure 5.4 Univariate analysis on tBodyAccMagmean

5.2 *Evaluation Metrics Used*

As the evaluation metrics confusion matrix and accuracy were used with the proposed model. So accuracy could be defined as ratio of total count of accurately classified points by count of total points.

$$\text{Accuracy} = \frac{\text{Total Number of correctly classified points}}{\text{Total Number of Points}}$$

Now accuracy alone can't be enough to describe the accuracy for the model hence, a confusion matrix, also known as an error matrix, [7] is a table layout for the performance visualization of an algorithm. In the confusion matrix row of the matrix represents the observations in a predicted class and column represents the observations in a true class [12]. Confusion matrix helps in checking whether the system is getting confused between two classes. Let say we have binary classification problem, with label 0 and 1, now confusion matrix is given as below:-

Predicted ↓ \ Actual →	0	1
0	True Negative	False Negative
1	False Positive	True Positive
	N	P

Figure 5.5 Confusion Matrix

Using the confusion matrix in Figure 5.5, we can define other evaluation metrics like precision, recall and f1-score.

Precision means of all the points the model predicted to be positive what percentage of them is actually positive. Recall is defined as of count of all points which actually belongs to class 1 how many out of them, model identifies as of class 1. We want all precision, recall and f1-score to be high [12].

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{12}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{13}$$

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{14}$$

5.3 *Results*

5.3.1 LSTM

Configuration of LSTM: With just single layer of LSTM, the model is trained with Dropout Rate of 0.233, 62 LSTM units, with activation as sigmoid, batch size 16 and count of epochs are set at 25. It gave 93.11% accuracy. It had total 18,234 parameters and took 161 KB space.

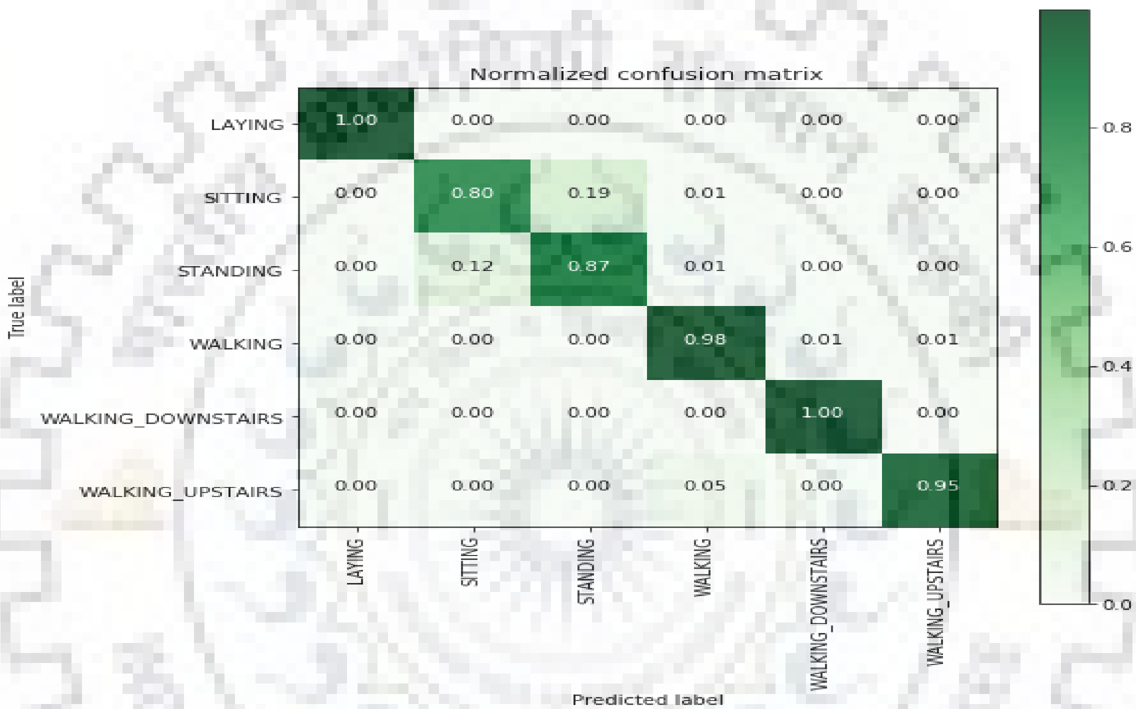


Figure 5.6 Confusion Matrix for LSTM

	Precision	Recall	F1-score	Support
LAYING	1.00	1.00	1.00	537
SITTING	0.86	0.80	0.83	491
STANDING	0.83	0.87	0.85	532
WALKING	0.94	0.98	0.96	496
WALKING_DOWNSTAIRS	0.99	1.00	1.00	420
WALKING_UPSTAIRS	0.99	0.95	0.97	471
Micro Average	0.93	0.93	0.93	2947
Macro Average	0.93	0.93	0.93	2947
Weighted Average	0.93	0.93	0.93	2947

Figure 5.7 Classification Report for LSTM

5.3.2 GRU

Configuration of GRU: With just single layer of GRU, the model is trained with Dropout Rate of 0.4, 50 GRU units, with activation as sigmoid, batch size 16 and count of epochs are set at 21. It gave 94.43% accuracy. It had total 9,306 parameters and took 91.7 KB space.

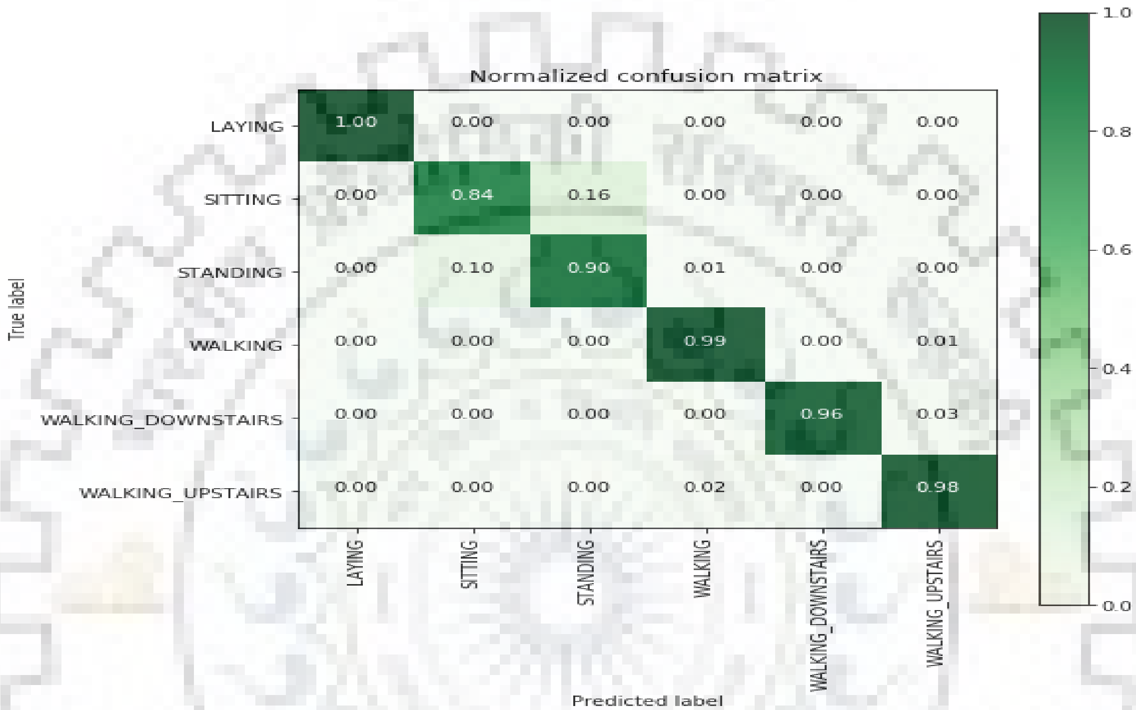


Figure 5.8 Confusion Matrix for GRU

	Precision	Recall	F1-score	Support
LAYING	1.00	1.00	1.00	537
SITTING	0.89	0.84	0.86	491
STANDING	0.86	0.90	0.88	532
WALKING	0.97	0.99	0.98	496
WALKING_DOWNSTAIRS	1.00	0.96	0.98	420
WALKING_UPSTAIRS	0.96	0.98	0.97	471
Micro Average	0.94	0.94	0.94	2947
Macro Average	0.94	0.94	0.94	2947
Weighted Average	0.94	0.94	0.94	2947

Figure 5.9 Classification Report for GRU

5.3.3 Bidirectional LSTM

Configuration of Bidirectional LSTM: With just single layer of Bidirectional LSTM, the model is trained with Dropout Rate of 0.233, 62 Bidirectional LSTM units, with activation as sigmoid, batch size 16 and count of epochs are set at 29. It gave 92.56% accuracy. It had total 36,462 parameters and took 307 KB space.

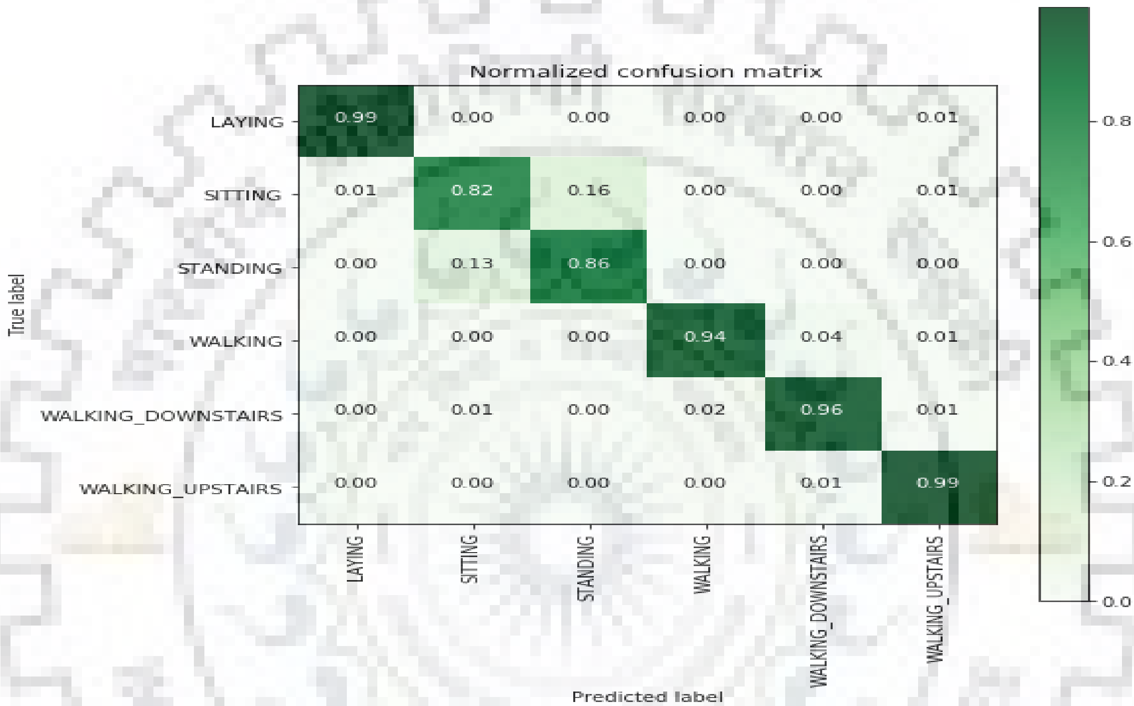


Figure 5.10 Confusion Matrix for Bidirectional LSTM

	Precision	Recall	F1-score	Support
LAYING	0.99	0.99	0.99	537
SITTING	0.84	0.82	0.83	491
STANDING	0.85	0.86	0.86	532
WALKING	0.98	0.94	0.96	496
WALKING_DOWNSTAIRS	0.94	0.96	0.95	420
WALKING_UPSTAIRS	0.95	0.99	0.97	471
Micro Average	0.93	0.93	0.93	2947
Macro Average	0.93	0.93	0.93	2947
Weighted Average	0.93	0.93	0.93	2947

Figure 5.11 Classification Report for Bidirectional LSTM

5.3.4 Bidirectional GRU

Configuration of Bidirectional GRU: With just single layer of Bidirectional GRU, the model is trained with Dropout Rate of 0.4, 50 Bidirectional GRU units, with activation as sigmoid, batch size 16 and count of epochs are set at 26. It gave 92.90% accuracy. It had total 18,606 parameters and took 167 KB space.

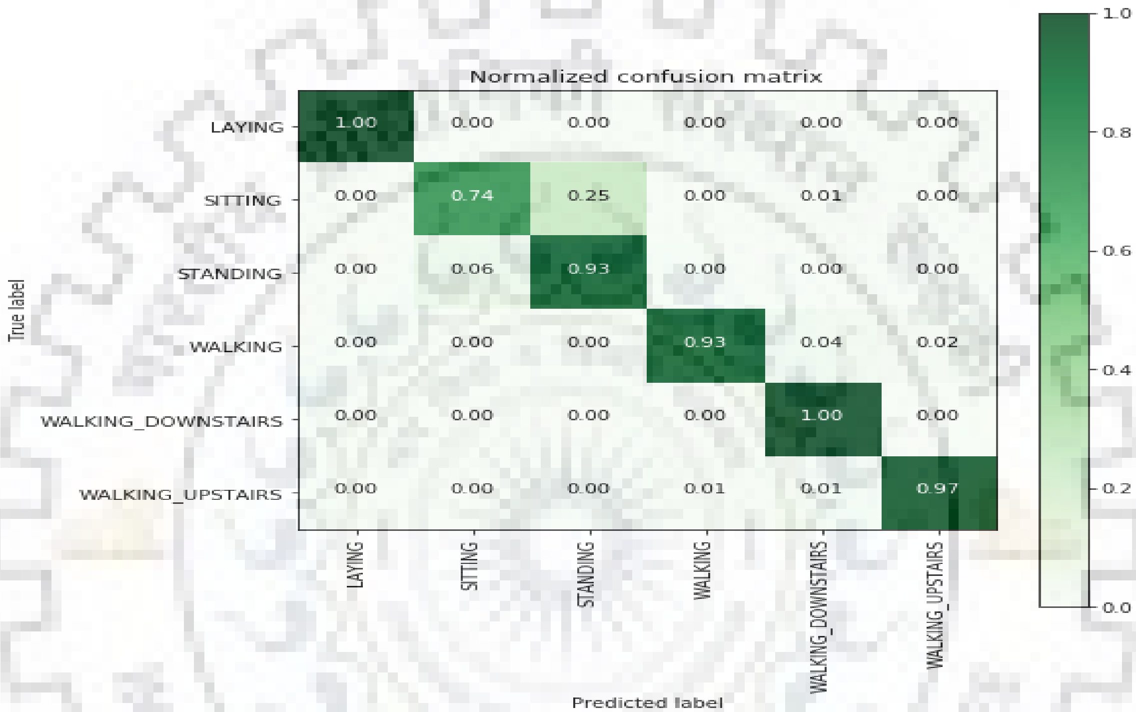


Figure 5.12 Confusion Matrix for Bidirectional GRU

	Precision	Recall	F1-score	Support
LAYING	1.00	1.00	1.00	537
SITTING	0.91	0.74	0.82	491
STANDING	0.80	0.93	0.86	532
WALKING	0.99	0.93	0.96	496
WALKING_DOWNSTAIRS	0.93	1.00	0.96	420
WALKING_UPSTAIRS	0.97	0.97	0.97	471
Micro Average	0.93	0.93	0.93	2947
Macro Average	0.93	0.93	0.93	2947
Weighted Average	0.93	0.93	0.93	2947

Figure 5.13 Classification Report for Bidirectional GRU

So, with raw data that is signal on all 3 axis converted into 128 size vector was provided as input to our RNN models and they gave pretty much good results. The comparison table for all models is mentioned below and the best model is highlighted:-

RNN models	Accuracy(%)
a) LSTM	93.11
b) GRU	94.43
c) Bidirectional LSTM	92.56
d) Bidirectional GRU	92.90
e) Ensemble(LSTM, GRU, Bidirectional LSTM and Bidirectional GRU) with MAX Voting	95.14

Table 5.1 Comparison of Accuracy of deep learning models

5.3.5 Logistic Regression

Configuration of Logistic Regression: This model using C value 30 and penalty as l2 gave the accuracy of 96.30%.

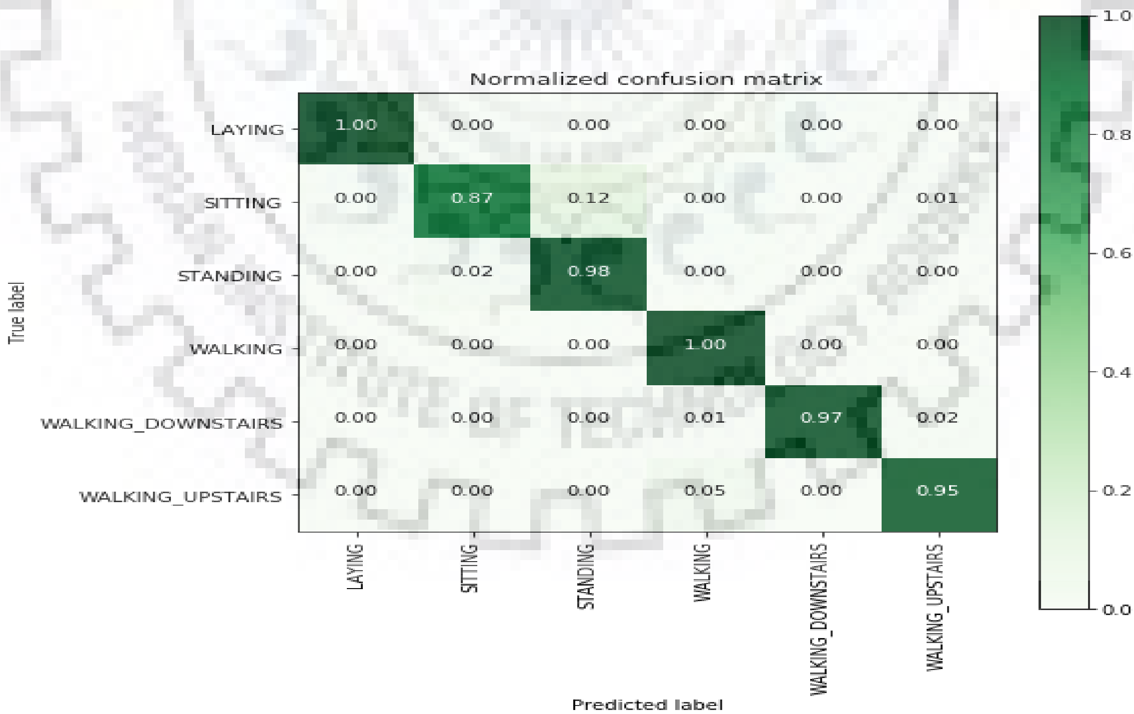


Figure 5.14 Confusion Matrix for Logistic Regression

	Precision	Recall	F1-score	Support
LAYING	1.00	1.00	1.00	537
SITTING	0.97	0.87	0.92	491
STANDING	0.90	0.98	0.94	532
WALKING	0.95	1.00	0.97	496
WALKING_DOWNSTAIRS	1.00	0.97	0.99	420
WALKING_UPSTAIRS	0.97	0.95	0.96	471
Micro Average	0.96	0.96	0.96	2947
Macro Average	0.97	0.96	0.96	2947
Weighted Average	0.96	0.96	0.96	2947

Figure 5.15 Classification Report for Logistic Regression

5.3.6 Linear SVC [1]

Configuration of Linear SVC: This model using C value 2 and tolerance as 0.00005 gave the accuracy of 96.50%.

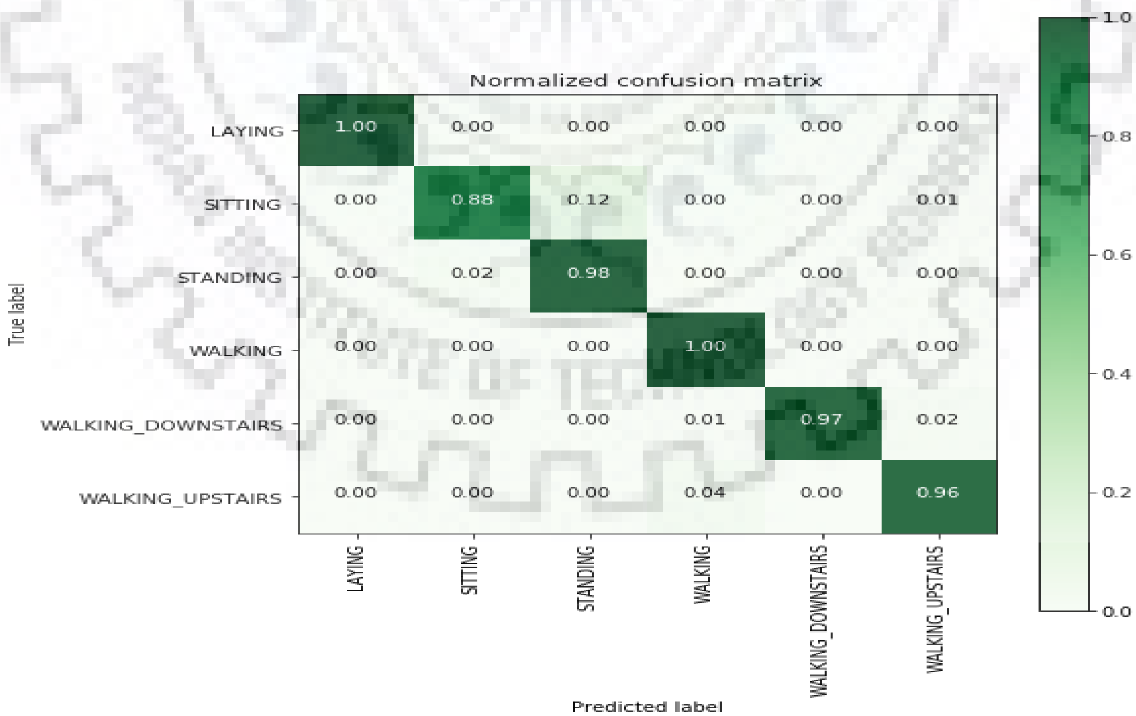


Figure 5.16 Confusion Matrix for Linear SVC

	Precision	Recall	F1-score	Support
LAYING	1.00	1.00	1.00	537
SITTING	0.98	0.87	0.92	491
STANDING	0.90	0.98	0.94	532
WALKING	0.96	1.00	0.98	496
WALKING_DOWNSTAIRS	1.00	0.97	0.98	420
WALKING_UPSTAIRS	0.97	0.96	0.97	471
Micro Average	0.97	0.97	0.97	2947
Macro Average	0.97	0.96	0.97	2947
Weighted Average	0.97	0.97	0.96	2947

Figure 5.17 Classification Report for Linear SVC

5.3.7 RBF SVC [1]

Configuration of RBF SVC: This model using C value 16 , kernel as RBF and gamma as 0.0078125 gave the accuracy of 96.02%.

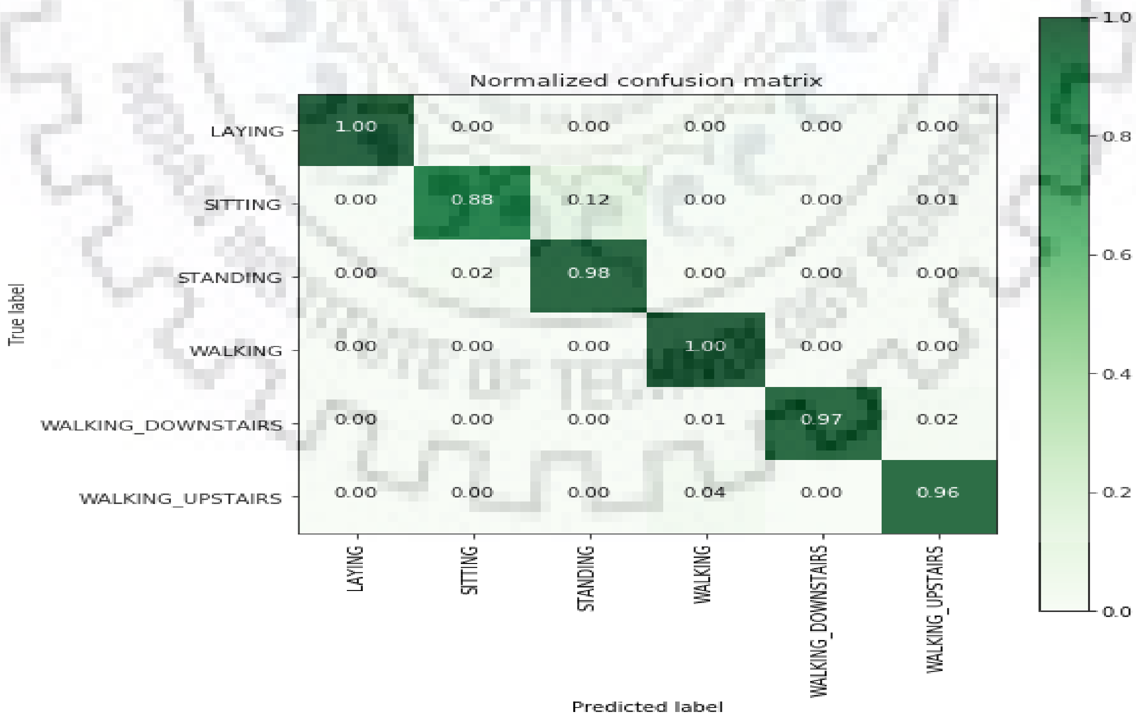


Figure 5.18 Confusion Matrix for RBF SVC

	Precision	Recall	F1-score	Support
LAYING	1.00	1.00	1.00	537
SITTING	0.97	0.91	0.94	491
STANDING	0.93	0.97	0.95	532
WALKING	0.95	0.98	0.96	496
WALKING_DOWNSTAIRS	0.98	0.93	0.96	420
WALKING_UPSTAIRS	0.94	0.96	0.95	471
Micro Average	0.96	0.96	0.96	2947
Macro Average	0.97	0.96	0.96	2947
Weighted Average	0.96	0.96	0.96	2947

Figure 5.19 Classification Report for RBF SVC

So, with all 561 engineered features was provided as input to our classical models and they gave pretty much good results. The comparison table for all models is mentioned below and the best model among them is highlighted:-

With Feature Engineering	%
a) Logistic Regression	96.30
b) Linear SVC [1]	96.50
c) RBF SVC [1]	96.02

Table 5.2 Comparison of Accuracy of classical machine learning models

5.3.8 Ensemble Method

The ensemble of best models was constructed. Results of Linear SVC, RBF SVC, Logistic Regression, LSTM and GRU were combined to give even better results. We used ensemble method with MAX Voting, by Sum Rule, by Product Rule and by Mean Rule.

5.3.8.1 *Ensemble Method with MAX Voting*

In this model if votes of two classes were found to be equal then as results of individual models were prioritized in order of accuracy, the first class with maximum votes is selected for the given input. This method worked with an accuracy of 96.97%.

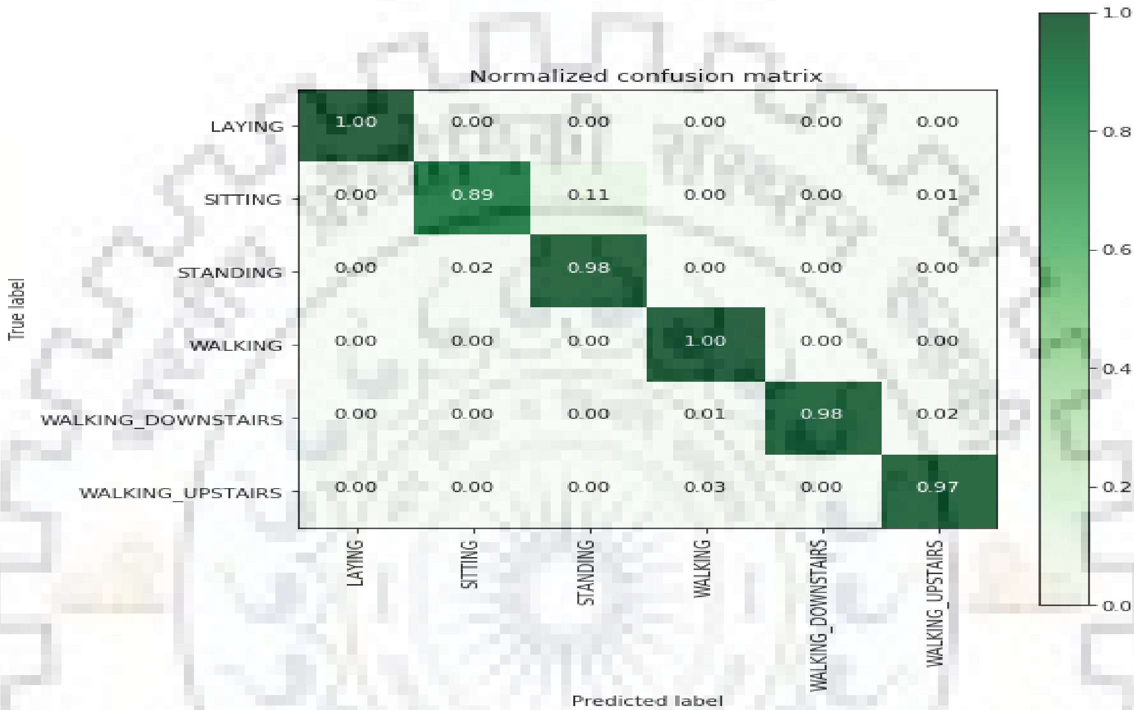


Figure 5.20 Confusion Matrix for MAX Votes Ensemble Technique

	Precision	Recall	F1-score	Support
LAYING	1.00	1.00	1.00	537
SITTING	0.98	0.89	0.93	491
STANDING	0.91	0.98	0.94	532
WALKING	0.97	1.00	0.98	496
WALKING_DOWNSTAIRS	1.00	0.98	0.99	420
WALKING_UPSTAIRS	0.98	0.97	0.98	471
Micro Average	0.97	0.97	0.97	2947
Macro Average	0.97	0.97	0.97	2947
Weighted Average	0.97	0.97	0.97	2947

Figure 5.21 Classification Report for MAX Votes Ensemble Technique

5.3.8.2 ***Ensemble Method with Sum Rule***

The accuracy of this model was found to be 97.35%.

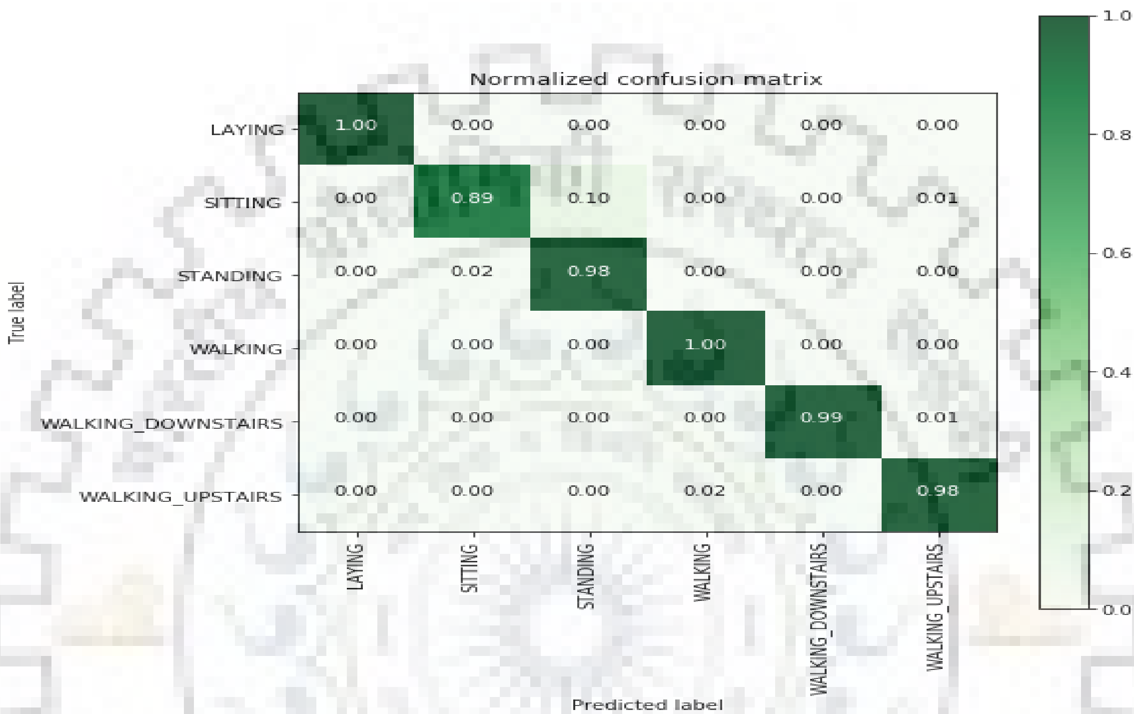


Figure 5.22 Confusion Matrix for Sum Rule Ensemble Technique

	Precision	Recall	F1-score	Support
LAYING	1.00	1.00	1.00	537
SITTING	0.98	0.89	0.93	491
STANDING	0.91	0.98	0.95	532
WALKING	0.98	1.00	0.99	496
WALKING_DOWNSTAIRS	1.00	0.99	0.99	420
WALKING_UPSTAIRS	0.99	0.98	0.99	471
Micro Average	0.97	0.97	0.97	2947
Macro Average	0.98	0.97	0.97	2947
Weighted Average	0.97	0.97	0.97	2947

Figure 5.23 Classification Report for Sum Rule Ensemble Technique

5.3.8.3 ***Ensemble Method with Product Rule***

The accuracy of this model was found to be 97.01%.

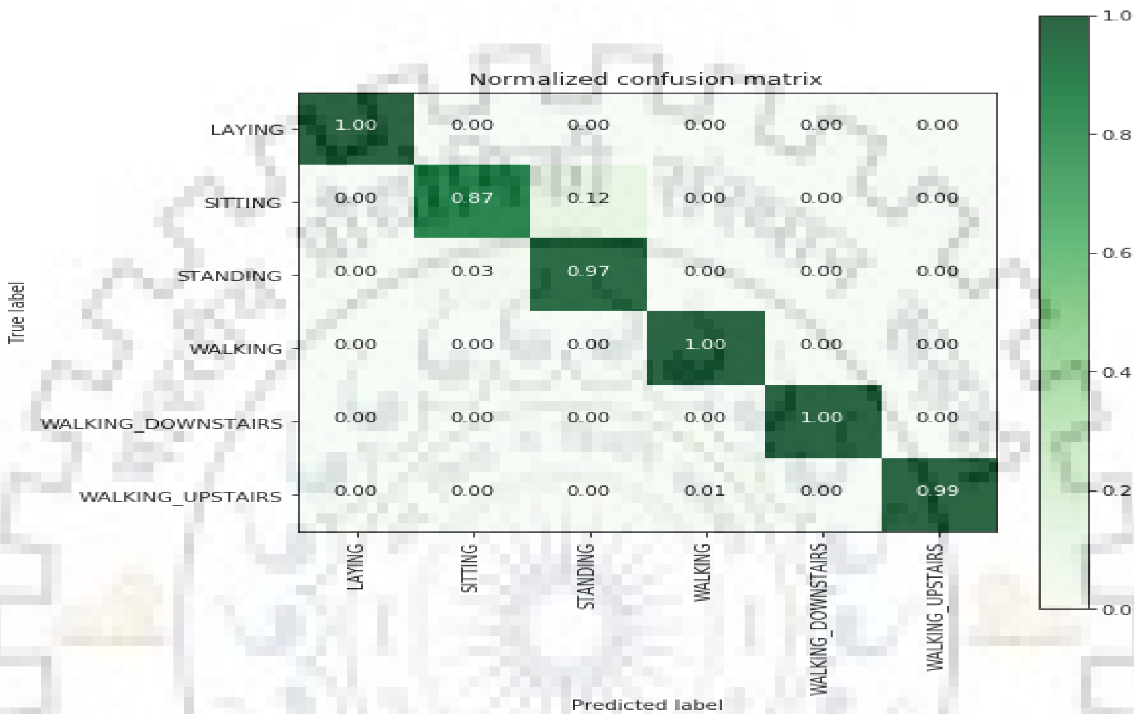


Figure 5.24 Confusion Matrix for Product Rule Ensemble Technique

	Precision	Recall	F1-score	Support
LAYING	1.00	1.00	1.00	537
SITTING	0.96	0.87	0.91	491
STANDING	0.89	0.97	0.93	532
WALKING	0.99	1.00	0.99	496
WALKING_DOWNSTAIRS	1.00	1.00	1.00	420
WALKING_UPSTAIRS	1.00	0.99	0.99	471
Micro Average	0.97	0.97	0.97	2947
Macro Average	0.97	0.97	0.97	2947
Weighted Average	0.97	0.97	0.97	2947

Figure 5.25 Classification Report for Product Rule Ensemble Technique

5.3.8.4 ***Ensemble Method with Mean Rule***

The accuracy of this model was found to be 97.35%.

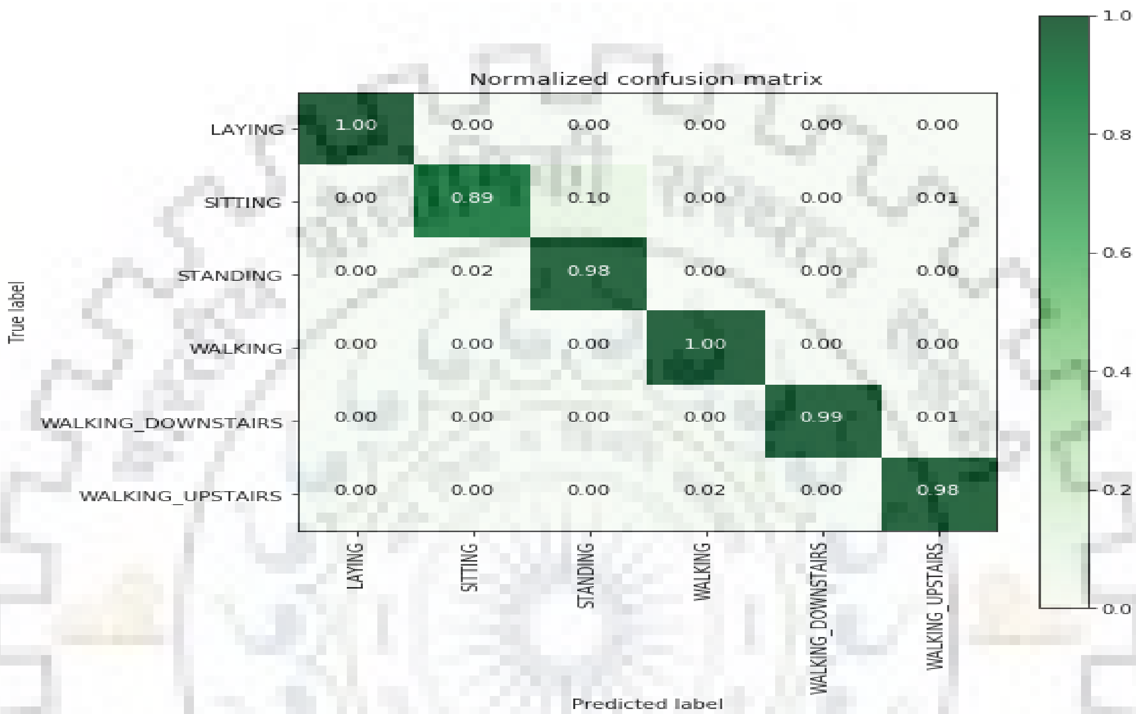


Figure 5.26 Confusion Matrix for Mean Rule Ensemble Technique

	Precision	Recall	F1-score	Support
LAYING	1.00	1.00	1.00	537
SITTING	0.98	0.89	0.93	491
STANDING	0.91	0.98	0.95	532
WALKING	0.98	1.00	0.99	496
WALKING_DOWNSTAIRS	1.00	0.99	0.99	420
WALKING_UPSTAIRS	0.99	0.98	0.99	471
Micro Average	0.97	0.97	0.97	2947
Macro Average	0.98	0.97	0.97	2947
Weighted Average	0.97	0.97	0.97	2947

Figure 5.27 Classification Report for Mean Rule Ensemble Technique

The model was cross checked on another dataset that is motion dataset [22] with 12 test point chosen (2 from each class) where it gave accuracy of 83.33%, that is 10 points out of 12 were correctly classified.

Finally, concluding with the final table with all accuracies in it and then selecting the best model among all the models available. In the below table all the ensemble model are designed using LSTM, GRU, Logistic Regression. Linear SVC and RBF SVC.

Technique	Accuracy (%)
LSTM	93.11
GRU	94.44
Bidirectional LSTM	92.56
Bidirectional GRU	92.91
Ensemble (LSTM, GRU, Bidirectional LSTM and Bidirectional GRU) with MAX Voting	95.14
Logistic Regression	96.30
Linear SVC [1]	96.50
RBF SVC [1]	96.02
Ensemble (LSTM, GRU, Logistic Regression. Linear SVC and RBF SVC) with MAX Voting	96.97
Ensemble (LSTM, GRU, Logistic Regression. Linear SVC and RBF SVC) with Sum Rule	97.35
Ensemble Technique with Product Rule	97.01
Ensemble (LSTM, GRU, Logistic Regression. Linear SVC and RBF SVC) with Mean Rule	97.35

Table 5.3 Comparison of All Techniques Implemented

Chapter 6 CONCLUSION AND FUTURE SCOPE

In this report and during my thesis we built various deep learning models and did ensemble learning. We performed the experiments on the data set [1] for Human action recognition. We built LSTM, Bidirectional LSTM, GRU and Bidirectional GRU and even an ensemble model of all these deep models by eliminating the overhead of feature engineering and got comparative results with the baseline model. The accuracy of baseline model was 96% [1] and we got 95.14% accuracy. Then to find the best model we took the problem much further and developed an ensemble model using the results of classical models and deep models and we got better and improved results than the existing model [1] that is 97.35% accuracy. As seen in the confusion matrix there was more misclassification between standing and sitting. So for that problem, incorporating barometer data could be useful. So further including that data might further improve the accuracy and we can even incorporate and train our model for some more human activities like jogging etc.

REFERENCES

[1]. Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz. A Public Domain Dataset for Human Activity Recognition Using Smartphones. 21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013. Bruges, Belgium 24-26 April 2013.

[2]. Butterworth, Stephen. "On the theory of filter amplifiers." Wireless Engineer 7, no. 6 (1930): 536-541.

[3]. Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9, no. 8 (1997): 1735-1780.

[4]. Cho, Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." arXiv preprint arXiv:1406.1078 (2014).

[5]. Gers, Felix A., Jürgen Schmidhuber, and Fred Cummins. "Learning to forget: Continual prediction with LSTM." (1999): 850-855.

[6]. Maaten, Laurens van der, and Geoffrey Hinton. "Visualizing data using t-SNE." Journal of machine learning research 9, no. Nov (2008): 2579-2605.

[7]. Stehman, Stephen V. "Selecting and interpreting measures of thematic classification accuracy." Remote sensing of Environment 62, no. 1 (1997): 77-89.

[8]. Siegelmann, Hava T., and Eduardo D. Sontag. "On the computational power of neural nets." Journal of computer and system sciences 50, no. 1 (1995): 132-150.

[9]. Hasan, S., Mohshi Masnad, H. Mahmud, and M. Hasan. "Human activity recognition using smartphone sensors with context filtering." In 9th International Conference on Advances in Computer-Human Interaction. ACHI, p. 69. 2016.

[10]. Cramer, Jan Salomon. "The origins of logistic regression." (2002): 4.

[11]. Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." Machine learning 20, no. 3 (1995): 273-297.

[12]. Powers, David Martin. "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation." (2011).

[13]. Schuster, Mike, and Kuldip K. Paliwal. "Bidirectional recurrent neural networks." IEEE Transactions on Signal Processing 45, no. 11 (1997): 2673-2681.

[14]. Opitz, David, and Richard Maclin. "Popular ensemble methods: An empirical study." Journal of artificial intelligence research 11 (1999): 169-198.

[15]. Schölkopf, Bernhard. "The kernel trick for distances." In Advances in neural information processing systems, pp. 301-307. 2001.

[16]. Theodoridis, Sergios, and Konstantinos Koutroumbas. "Pattern recognition. 2003." Elsevier Inc (2009).

[17]. Chung, Junyoung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. "Empirical evaluation of gated recurrent neural networks on sequence modeling." arXiv preprint arXiv:1412.3555 (2014).

[18]. Britz, Denny. "Recurrent neural network tutorial, part 4-Implementing a GRU/LSTM RNN with Python and Theano." URL http://www. wildml. com/2015/10/recurrent-neural-network-tutorial-part-4-implementing-a-grulstm-rnn-with-python-and-theano (2015).

[19]. Polikar, Robi. "Ensemble based systems in decision making." IEEE Circuits and systems magazine 6, no. 3 (2006): 21-45.

[20]. Rokach, Lior. "Ensemble-based classifiers." Artificial Intelligence Review 33, no. 1-2 (2010): 1-39.

[21]. Petneházi, Gábor. "Recurrent neural networks for time series forecasting." arXiv preprint arXiv:1901.00069 (2019).

[22]. Bartlett, James, Vinay Prabhu, and John Whaley. "Acctionnet: A dataset of human activity recognition using on-phone motion sensors." In Proceedings of the 34th International Conference on Machine Learning (Sydney, Australia, 2017).