Time Series Analysis

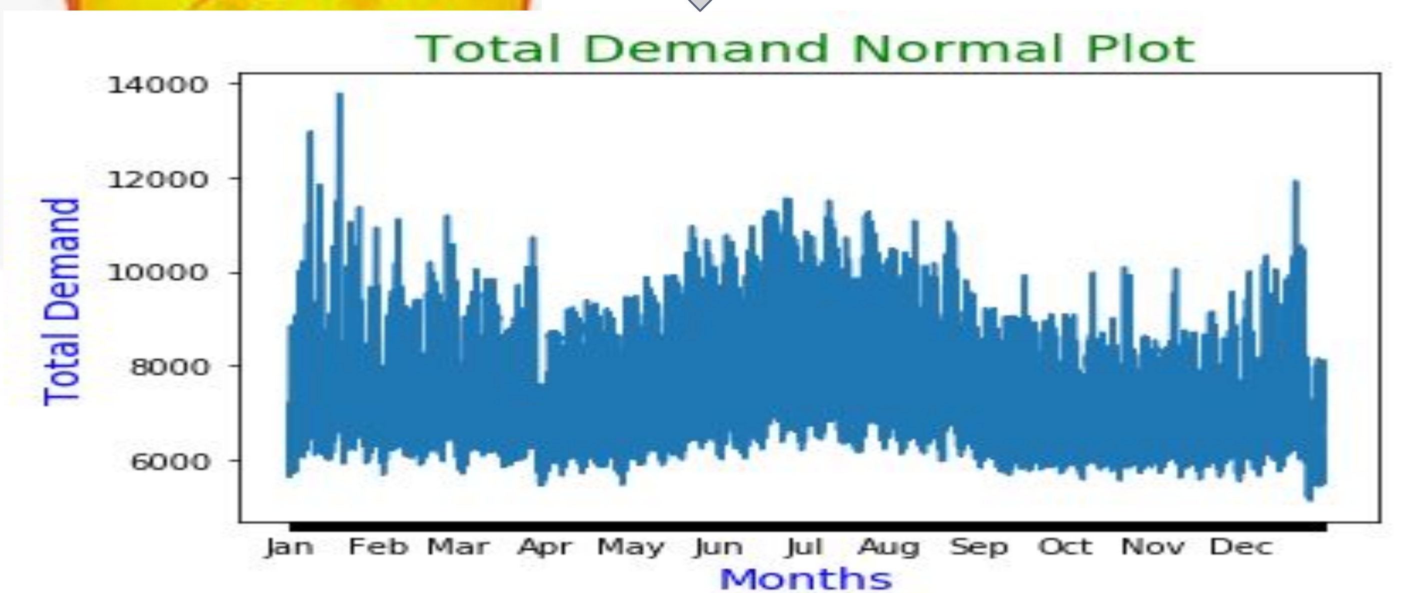
Jimson Mathew, Dept. CSE, IIT P

What is Time Series? / Types of Time Series?

by a

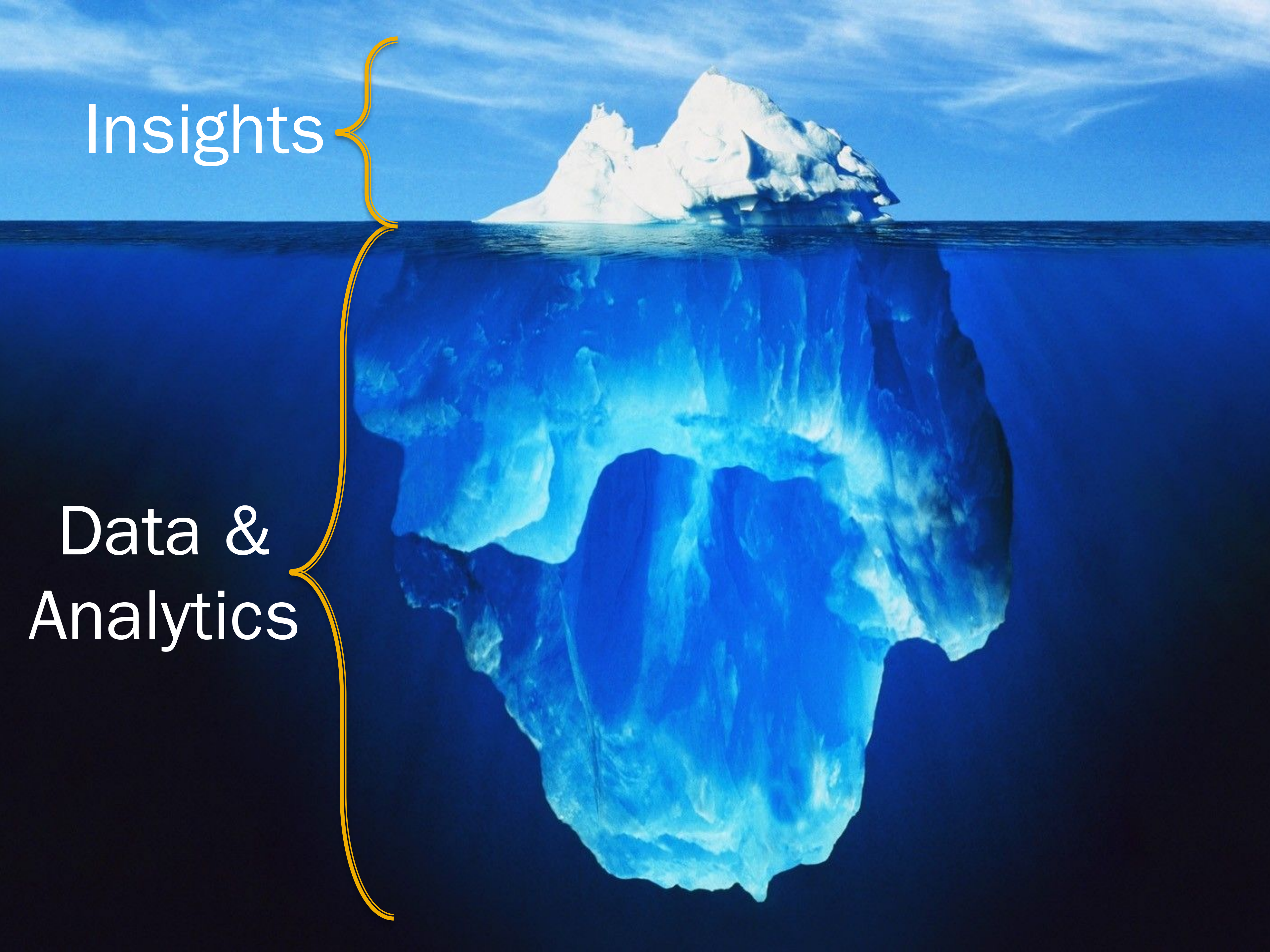
e: time series
single observation
recorded over regular
interval of time

month...)



Time Series Forecasting Models

- ❑ Autoregressive Models (AR)
- ❑ Moving Average Models (MA)
- ❑ Autoregressive Moving Average Models (ARMA)
- ❑ Autoregressive Integrated Moving Average Models (ARIMA)
- ❑ Neural Network models
- ❑ Long Short Term Memory Network Models (LSTM)


Insights
Data & Analytics

Smart Grid

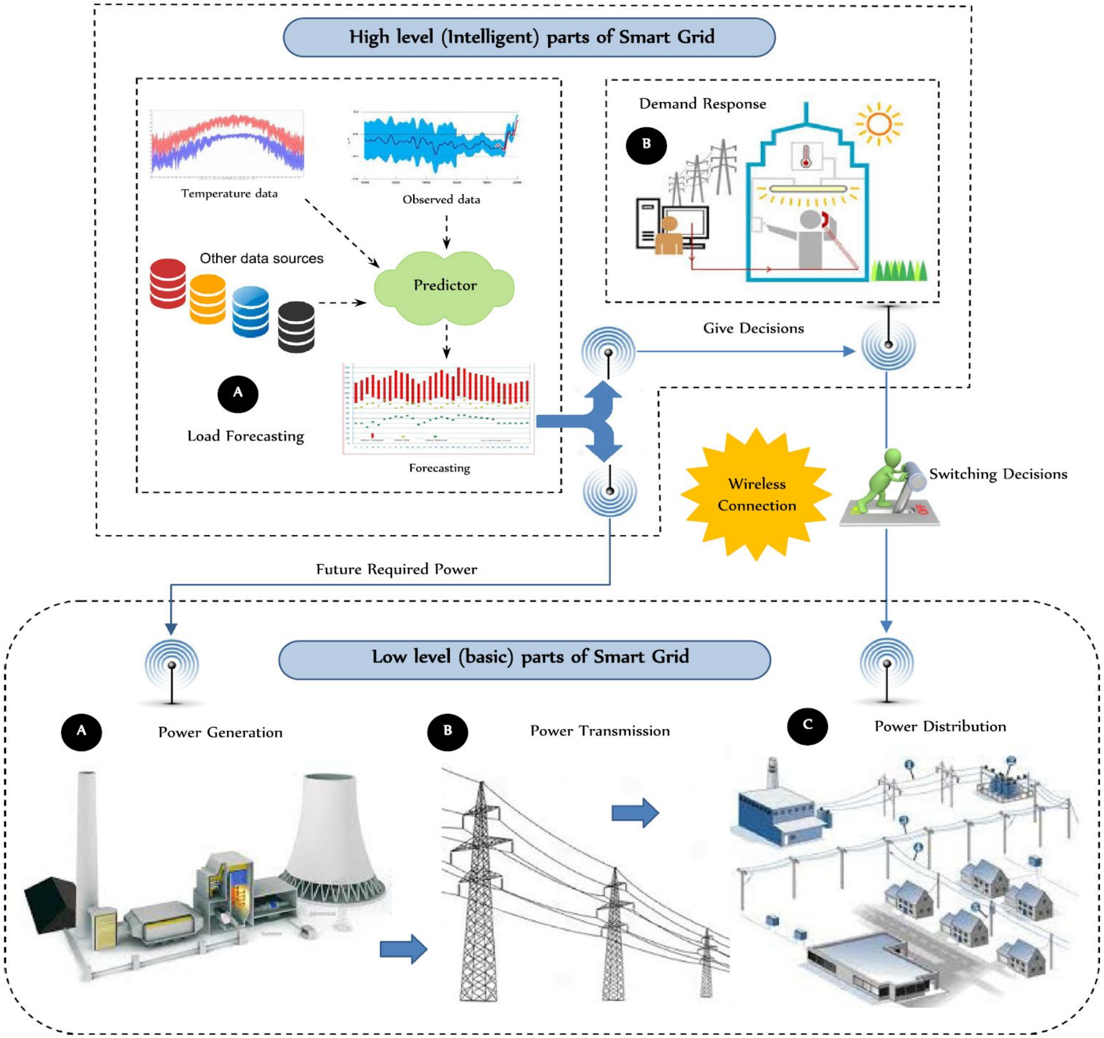


Fig. 1. Architecture and main components of smart grid.

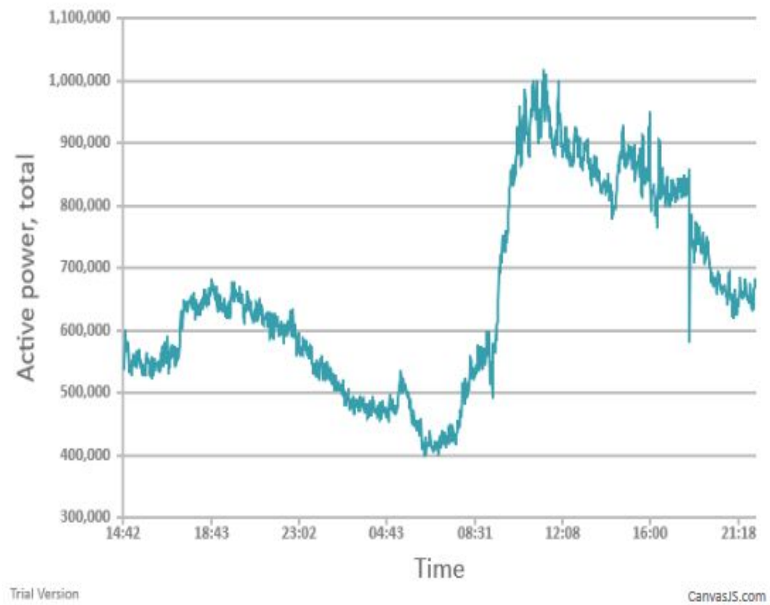
Architecture and main components of smart grid.

Eg:Real Time Electricity Usage at IITP

24 Hour Data Data by Date

Show All entries Search:

Date (dd-mm-yyyy)	Time (hh:mm:ss)	Readings (In standard unit)
06-11-2017	21:59:06	676170.38
06-11-2017	21:58:06	682617.75
06-11-2017	21:57:06	672862.81
06-11-2017	21:56:06	669819.19
06-11-2017	21:55:06	666511.94
06-11-2017	21:54:06	633467.06
06-11-2017	21:53:06	655203.06
06-11-2017	21:52:06	649298.5
06-11-2017	21:51:06	643586.75
06-11-2017	21:50:06	629989.38
06-11-2017	21:49:06	640710.94



Eg:Real Time Electric Energy Usage at IITP

24 Hour Data | **Data by Date**

Filter by date 15-10-2017 to 06-11-2017 Proceed

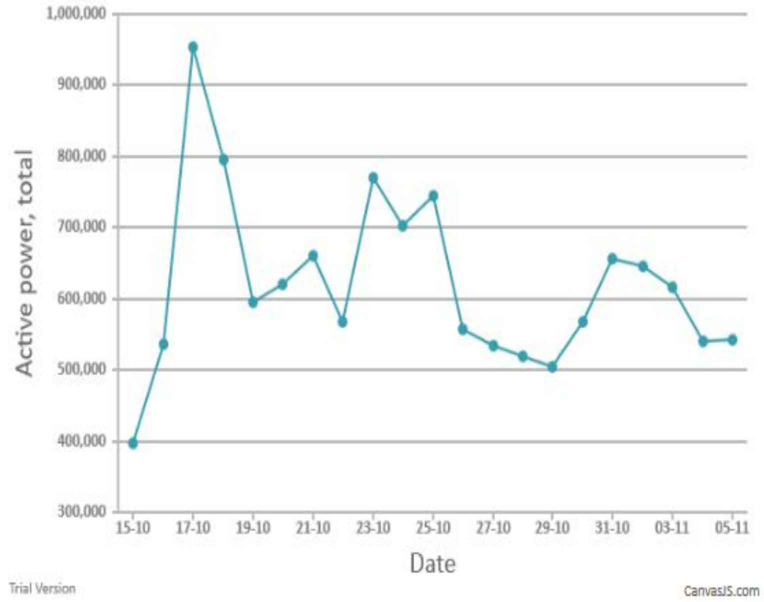
Show All entries

Search:

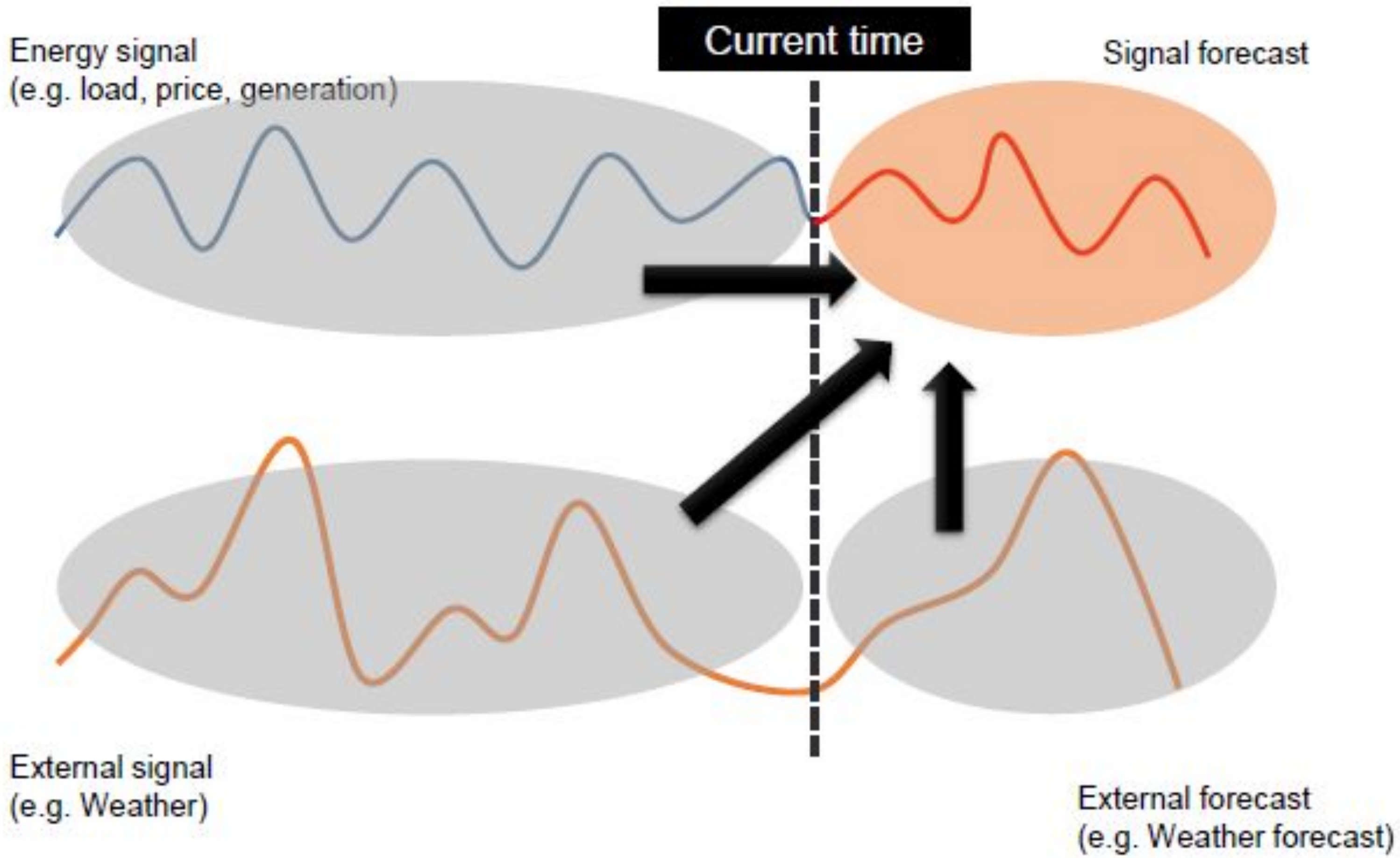
Date	Average Reading
05-11-2017	541868.95
04-11-2017	540414.56
03-11-2017	616917.99
02-11-2017	645896.68
31-10-2017	656820.58
30-10-2017	567431.32
29-10-2017	504964.03
28-10-2017	519225.13
27-10-2017	534465.36
26-10-2017	556953.28
25-10-2017	744588.47

Showing 1 to 21 of 21 entries

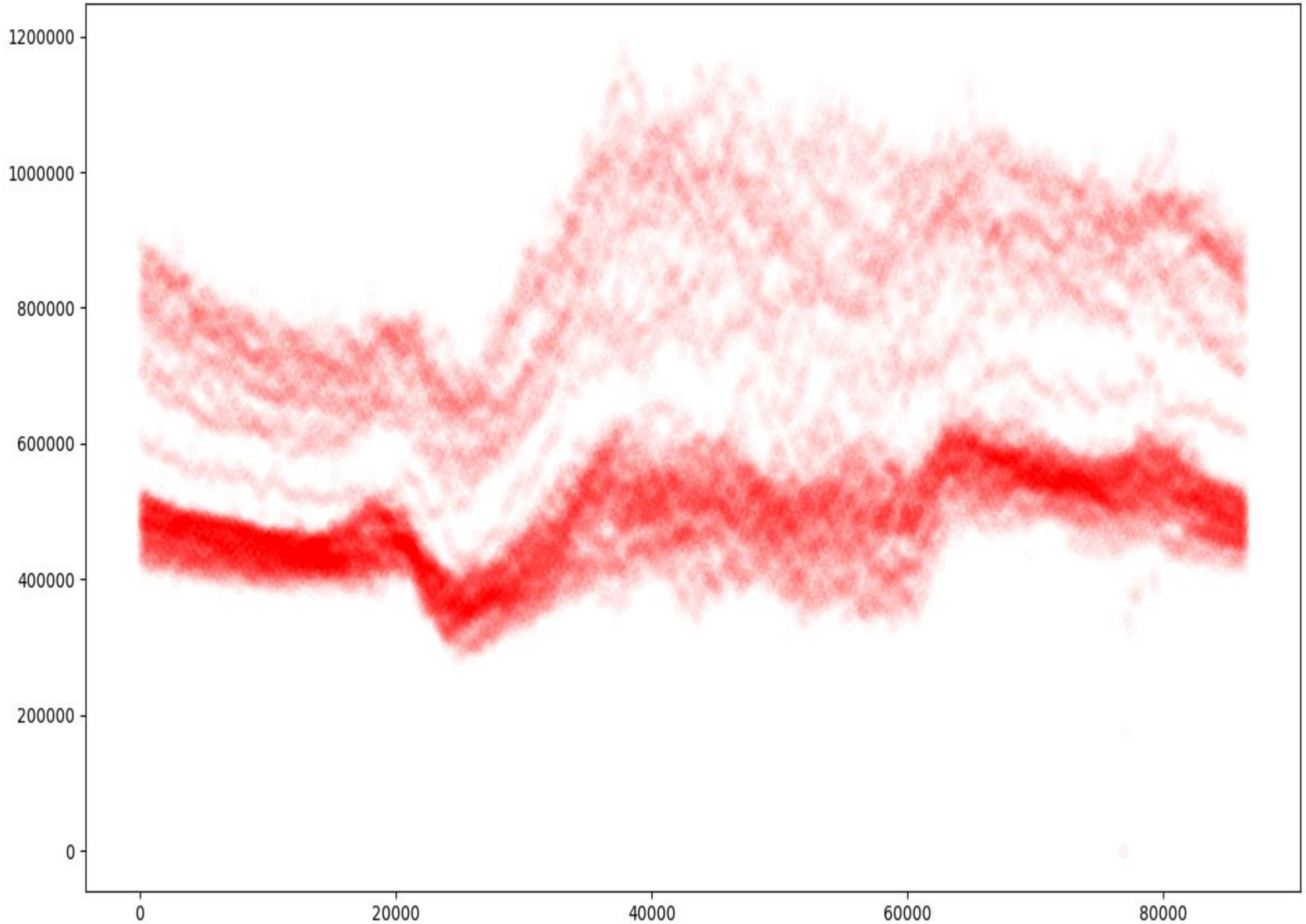
Previous 1 Next



Energy Forecast



Eg:Real Time Electric Energy Usage at IITP



Autoregressive Models (AR)

- Values of variables in one period related to its value in previous periods
- AR (p): AR model with p lags

AR(p): $$y_t = \sum_{1}^{p} \Theta_i . y_{t-i} + \epsilon_t$$

AR(1): $$y_t = \Theta_1 . y_{t-1} + \epsilon_t$$

where

- ϵ_t is zero mean uncorrelated random variables
- θ_i are autoregressive coefficients (parameter)
- Yt is observed variable (predicted)

Moving Average Models

- Relationship between a variable and residuals from the previous lags
- MA(q): moving average with q lags

MA(p): $$y_t = \sum_{1}^{p} \Theta_i . \epsilon_{t-i} + \epsilon_t$$

MA(1): $$y_t = \Theta_1 . \epsilon_{t-1} + \epsilon_t$$

where

- are the moving average coefficients (parameters)

Autoregressive Moving Average Models

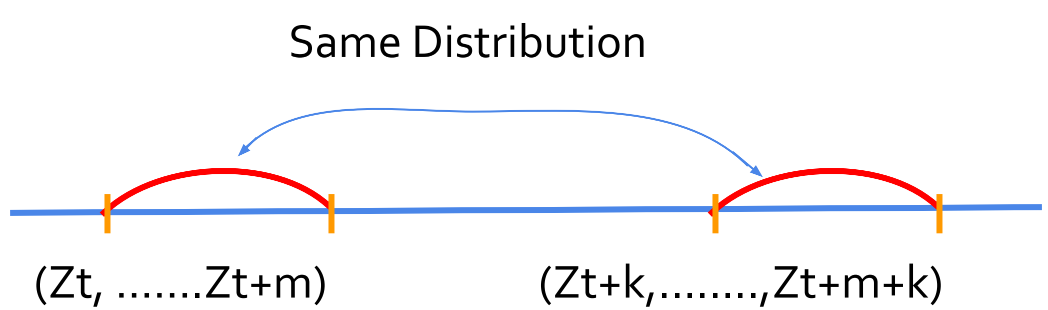
- Combine both ‘p’ autoregressive terms and ‘q’ moving average terms i.e   ARMA(p,q)

$$y_t = \mu + \sum_{1=1}^{p} y_i . y_{t-i} + \epsilon_t + \sum_{i=1}^{q} \theta_i . \epsilon_{t-i}$$

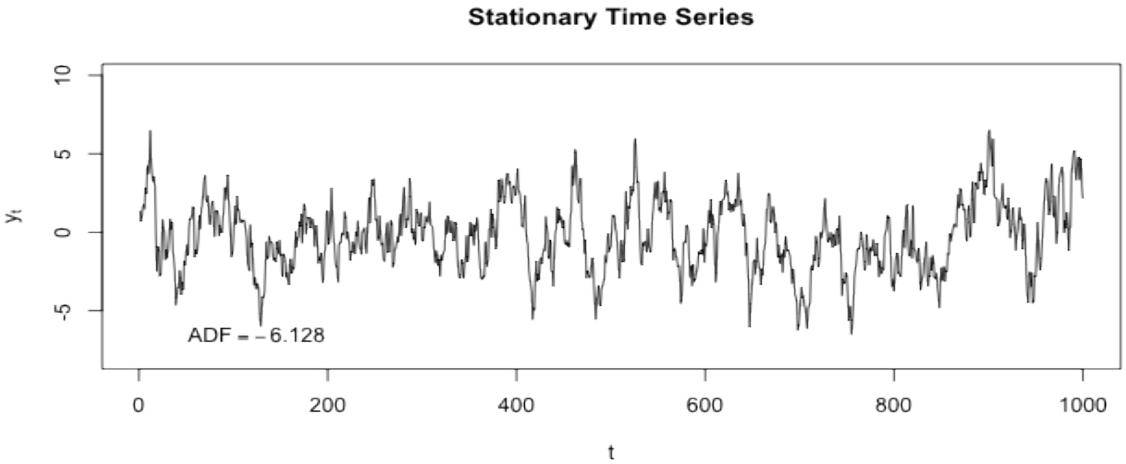
- modelling AR, MA, ARMA(p,q) requires **stationary process**
- mean and variance has to be constant
- process shouldn’t have trend

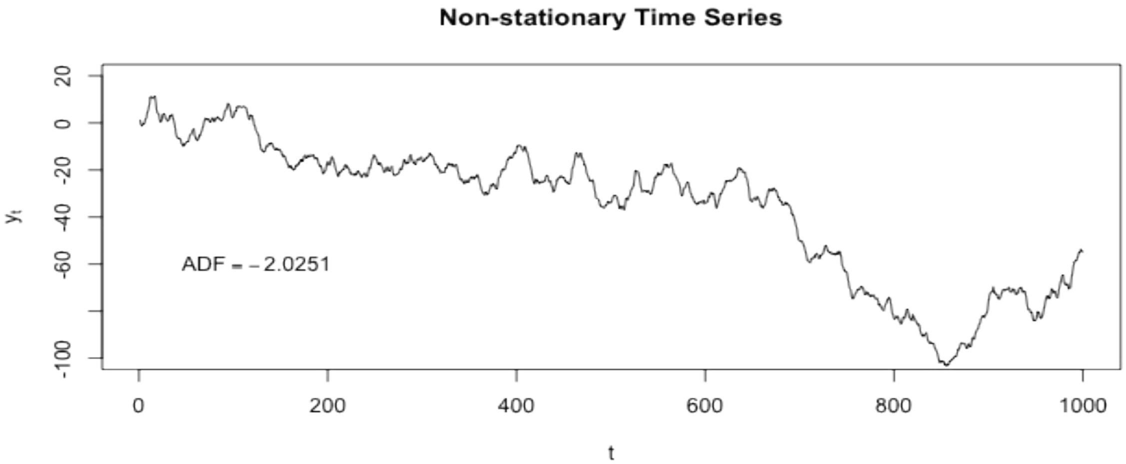
Stationarity

- A sequence of random variables Z = {Zt} belongs to real numbers is **stationary** if its distribution is invariant to shifting in time
- a process having zero mean and constant variance is a stationary process



Examples of stationary and non-stationary series





Autoregressive Integrated Moving Average models (ARIMA) continued

- ARIMA model can be represented as ARIMA(p,d,q)
  - **p** is the number of autoregressive terms
  - **d** is the number of non seasonal differences needed for stationarity
  - **q** is the number of lagged forecast errors in the prediction equation
- The forecasting equation is constructed as follows
  - If d=0: $y_t = Y_t$
  - If d=1: $y_t = Y_t - Y_{t-1}$
  - If d=2: $y_t = (Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2}) = Y_t - 2Y_{t-1} + Y_{t-2}$

Autoregressive Integrated Moving Average models (ARIMA) continued

- Identifying the order of differencing in an ARIMA model
  - **Rule 1:** If the series has positive autocorrelations out to a high number of lags, then it probably needs a higher order of differencing
  - **Rule 2:** If the lag-1 autocorrelation is zero or negative, or the autocorrelations are all small and pattern less, then the series does not need a higher order of differencing
  - **Rule 3:** The optimal order of differencing is often the order of differencing at which the standard deviation is lowest

Multiple Linear Regression models (MLR)

- A linear regression model containing more than one predictor variable
- It is used to analyze the association between two or more independent variables and a single dependent variable

$$\hat{Y} = b_0 + b_1x_1 + b_2x_2 + ... + b_px_p$$

$\hat{Y}$ = predicted value of dependent variable

x1 through xp are distinct independent variables

bo= is the value of Y when all the independent variable are zero

b1 through bp are regression coefficients

Feedforward Neural Network

- The information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output node
  - Input to the network is n-dimensional
  - The network contains L-1 hidden (2, in this case) having n neurons each
  - Finally, one output layer containing k neurons
  - Each neuron in hidden and and output layer can be split into two parts:
    - **pre-activation**
    - **activation**

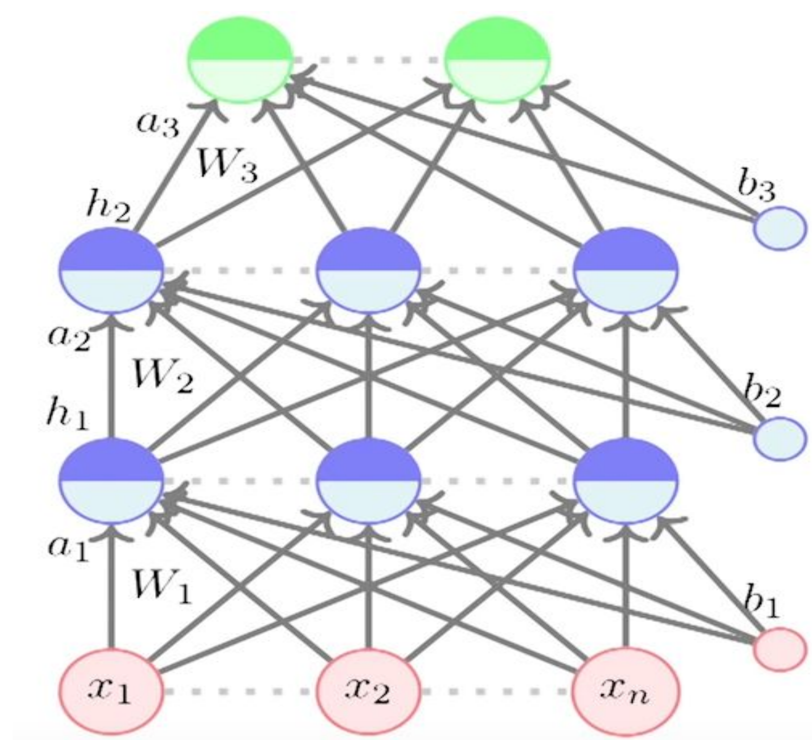
- The pre-activation at layer i is given by: $a_i(x) = b_i + W_i h_{i-1}(x)$
- The activation at layer i is given by: $h_i(x) = g(a_i(x))$

where g is called the activation function(for example, logistic, tanh, linear, etc)

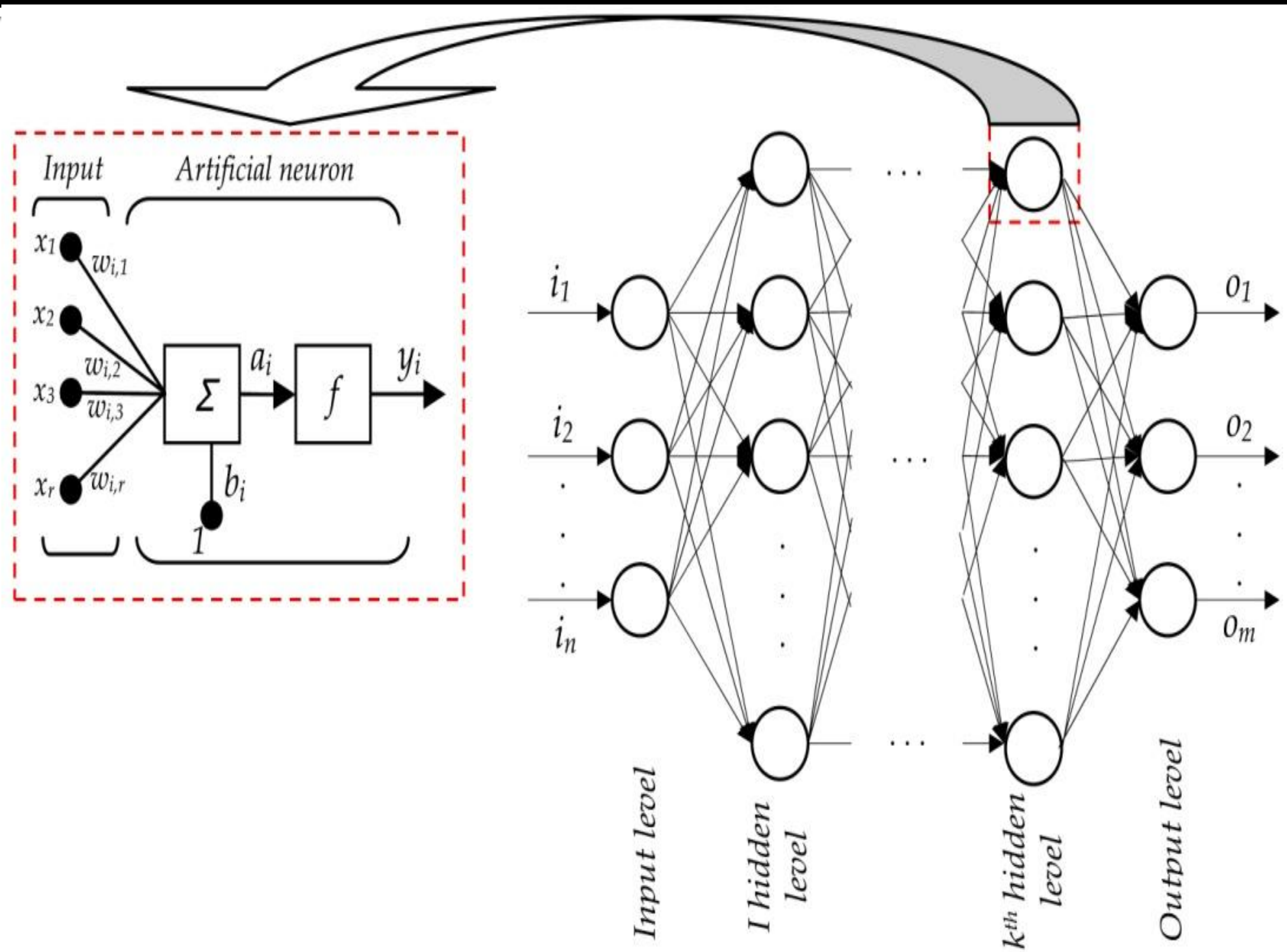
- The activation at the output layer i is given by: $f(x) = h_L(x) = O(a_L(x))$

where O is the output activation function (for example, softmax, linear, etc)

- **Model:** $y_i = f(x_i) = O(W^3 g(W^2 g(W^1 x + b_1) + b_2) + b_3)$



Conventional Deep learning

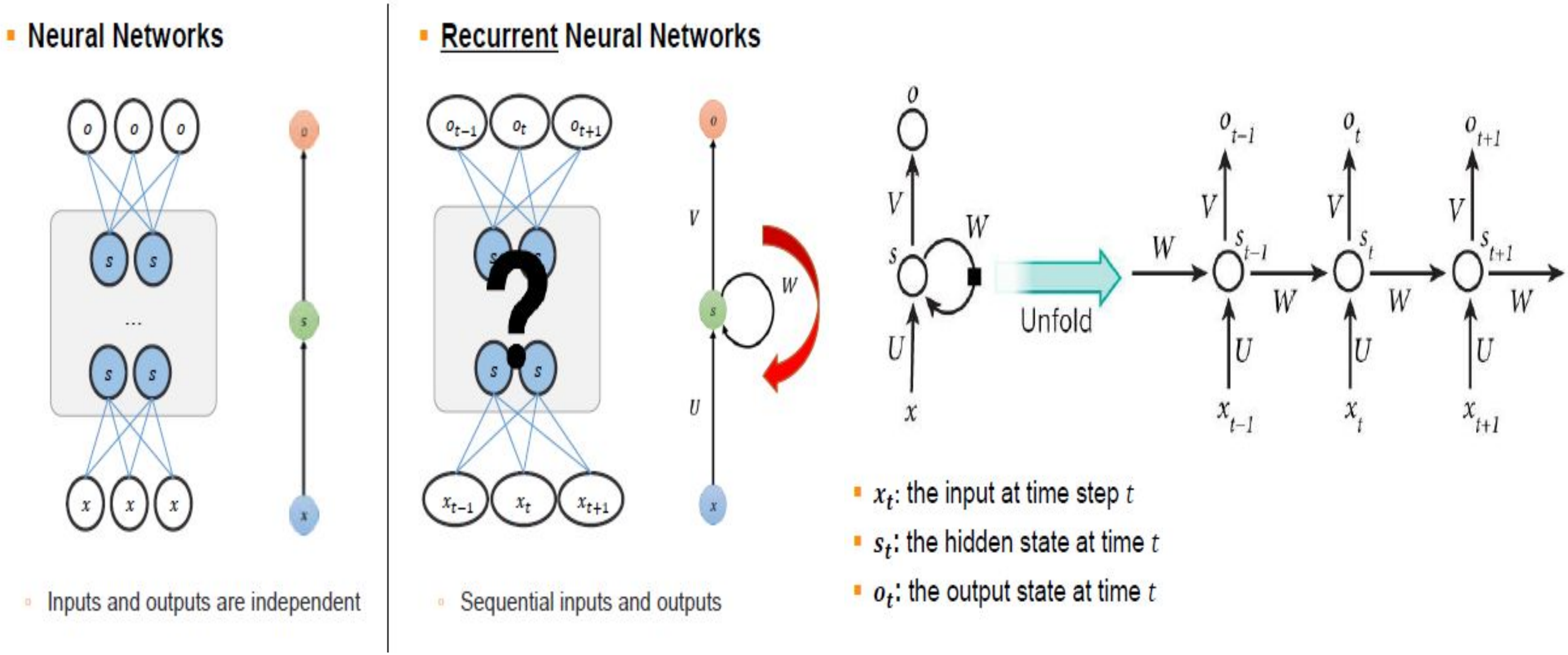


Aarshay Jain

https://www.analyticsvidhya.com/blog/2016/03/introduction-deep-learning-fundamentals-neural-networks/

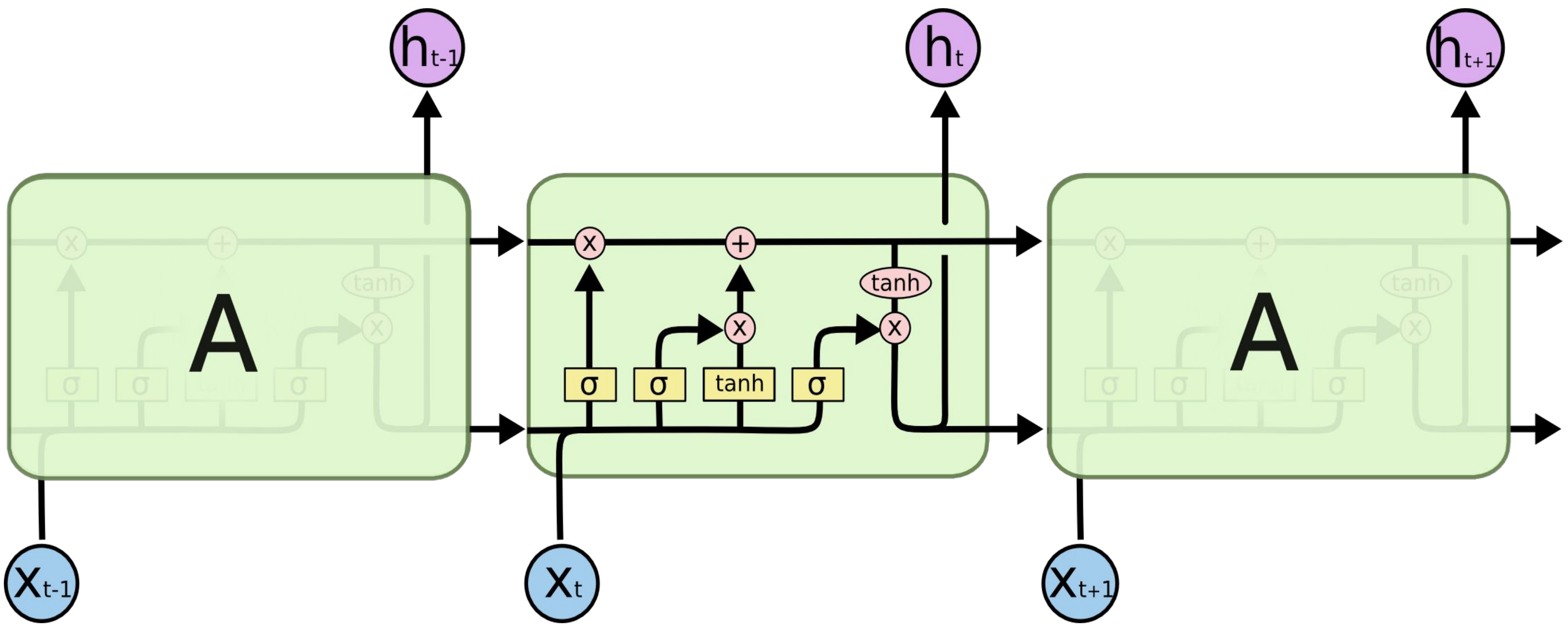
Recurrent Neural Networks

Time series forecasting with Recurrent Neural Networks (RNN)

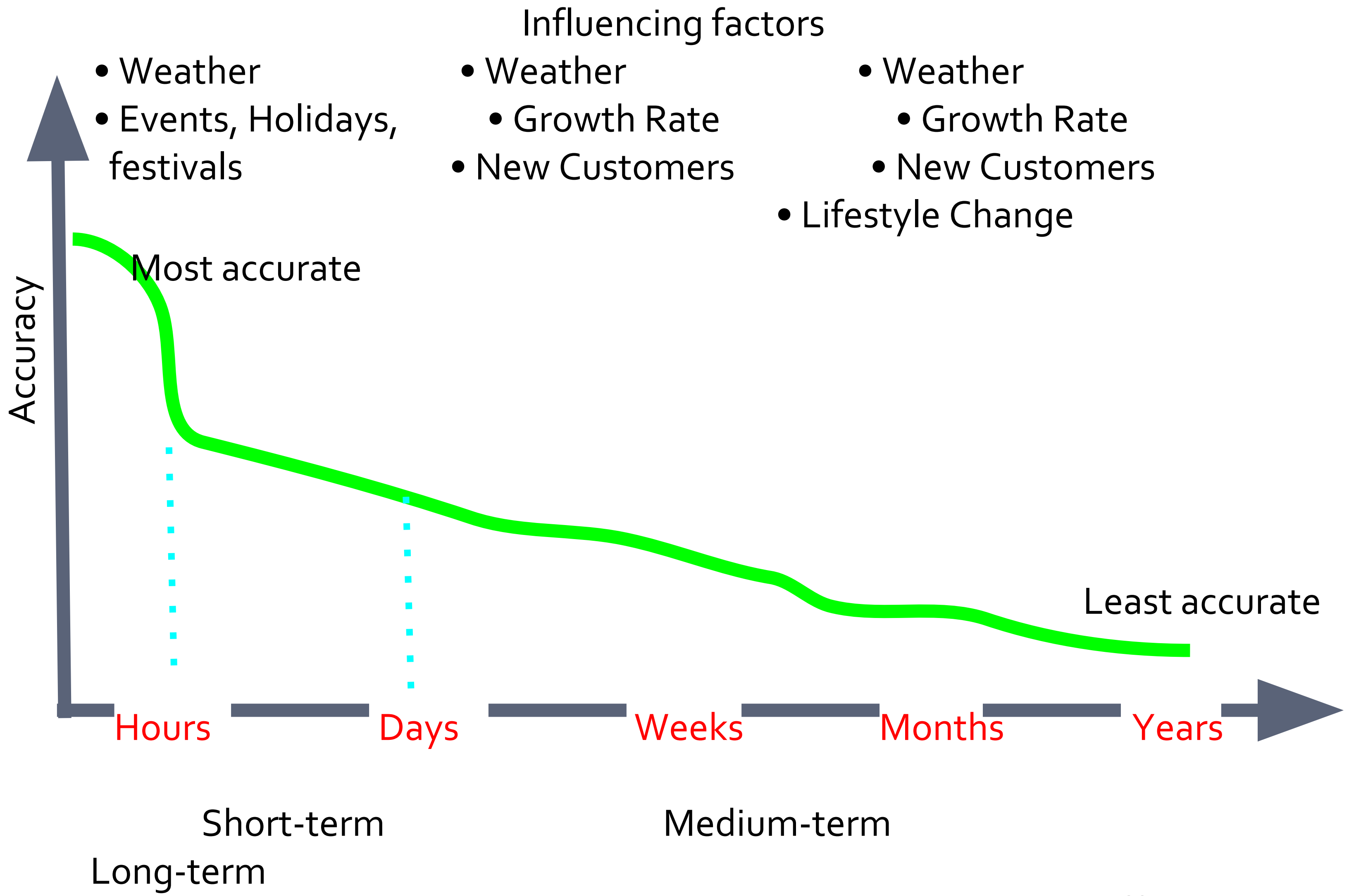


LSTM Networks

- Special kind of RNNs capable of learning long term dependencies
- Four neural network layer,, interacting in a very special way



Load Forecasting Accuracy



Forecast results

Forecast results for Kaggle electrical load dataset:

Time Of Year (Seasons)	RMSE					MAPE				
	SVR	FNN	DBN	LSTM-SSA-MSE	LSTM-SSA-PLF	SVR	FNN	DBN	LSTM-SSA-MSE	LSTM-SSA-PLF
Spring	59044.76	62939.39	62658.25	59711.83	59525.31	76.38%	79.83%	80.32%	73.06%	74.38%
Summer	61617.91	62074.01	60981.38	66204.35	66273.94	91.39%	81.66%	72.35%	75.48%	71.06%
Autumn	63728.59	64483.76	62622.23	59706.26	56335.94	65.47%	58.15%	60.46%	63.31%	62.17%
Winter	64662.19	62532.44	66387.25	58066.71	60929.47	69.26%	75.71%	73.23%	70.26%	69.86%
Average	62263.36	63007.40	63162.28	60922.28	**60766.16**	75.62%	73.84%	71.34%	70.50%	**69.36%**

Forecast results for South Australia electrical load dataset:

Time Of Year (Seasons)	RMSE					MAPE				
	SVR	FNN	DBN	LSTM-SSA-MSE	LSTM-SSA-PLF	SVR	FNN	DBN	LSTM-SSA-MSE	LSTM-SSA-PLF
Spring	38.66	36.36	37.81	30.67	28.14	1.54%	2.26%	2.55%	1.48%	1.56%
Summer	40.31	35.77	35.46	20.39	20.67	1.98%	1.79%	1.75%	1.41%	1.22%
Autumn	45.57	45.92	54.15	29.04	30.70	2.63%	1.55%	2.07%	2.02%	1.84%
Winter	30.77	29.97	37.25	32.27	32.25	1.99%	2.09%	2.18%	1.65%	1.75%
Average	38.82	37.00	41.16	28.09	27.94	2.03%	1.92%	2.13%	1.64%	1.59%

Forecast results for Tasmania electrical load dataset:

Time Of Year (Seasons)	RMSE					MAPE				
	SVR	FNN	DBN	LSTM-SSA-MSE	LSTM-SSA-PLF	SVR	FNN	DBN	LSTM-SSA-MSE	LSTM-SSA-PLF
Spring	18.67	19.61	24.65	15.24	13.91	1.25%	1.32%	1.78%	1.21%	1.17%
Summer	22.76	21.93	25.87	13.81	15.67	1.28%	1.31%	1.77%	1.07%	1.26%
Autumn	24.22	23.40	22.59	15.39	16.11	1.59%	1.58%	1.51%	1.57%	1.41%
Winter	22.19	22.66	20.99	22.17	22.93	1.42%	1.41%	1.30%	1.31%	1.10%
Average	21.96	21.90	23.52	16.65	17.15	1.38%	1.40%	1.56%	1.29%	1.24%

Forecast results for Queensland electrical load dataset:

Time Of Year (Seasons)	RMSE					MAPE				
	SVR	FNN	DBN	LSTM-SSA-MSE	LSTM-SSA-PLF	SVR	FNN	DBN	LSTM-SSA-MSE	LSTM-SSA-PLF
Spring	59.66	60.29	82.34	58.13	59.86	0.75%	0.72%	1.01%	0.72%	0.73%
Summer	56.38	68.79	70.49	57.42	57.73	0.72%	0.87%	0.91%	0.67%	0.67%
Autumn	58.61	52.10	67.03	57.39	51.37	0.82%	0.71%	0.99%	0.68%	0.66%
Winter	50.16	55.50	56.96	51.20	49.76	0.67%	0.73%	0.74%	0.66%	0.65%
Average	56.20	59.17	69.20	56.35	54.68	0.74%	0.75%	0.91%	0.68%	0.68%

Forecast results for Victoria electrical load dataset:

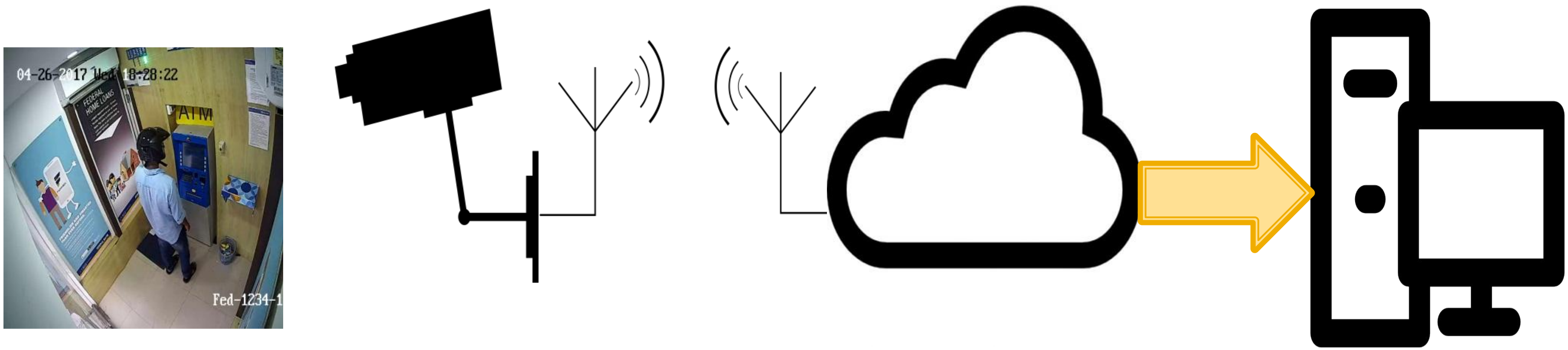
Time Of Year (Seasons)	RMSE					MAPE				
	SVR	FNN	DBN	LSTM-SSA-MSE	LSTM-SSA-PLF	SVR	FNN	DBN	LSTM-SSA-MSE	LSTM-SSA-PLF
Spring	67.32	76.63	73.97	66.29	67.14	1.06%	1.26%	1.39%	0.99%	0.89%
Summer	67.19	77.12	85.57	70.21	70.51	1.06%	1.21%	1.22%	0.89%	0.93%
Autumn	86.74	90.89	122.66	72.83	72.33	1.17%	1.23%	1.73%	1.03%	0.97%
Winter	63.17	79.50	69.99	73.48	72.21	0.86%	1.17%	1.06%	0.99%	1.06%
Average	71.11	81.03	88.47	70.45	70.45	1.04%	1.22%	1.35%	0.97%	0.96%

Forecast results for New South Wales electrical load dataset:

Time Of Year (Seasons)	RMSE					MAPE				
	SVR	FNN	DBN	LSTM-SSA-MSE	LSTM-SSA-PLF	SVR	FNN	DBN	LSTM-SSA-MSE	LSTM-SSA-PLF
Spring	88.13	81.62	97.01	87.57	86.27	0.88%	0.78%	1.05%	0.86%	0.86%
Summer	81.50	85.57	72.25	78.31	80.18	0.82%	0.89%	0.75%	0.80%	0.81%
Autumn	67.36	112.44	89.55	71.16	73.87	0.64%	1.06%	0.87%	0.71%	0.70%
Winter	71.24	84.59	91.31	81.06	78.13	0.63%	0.80%	0.80%	0.69%	0.70%
Average	77.62	91.05	87.53	79.52	79.61	0.74%	0.88%	0.86%	0.76%	0.77%

Model	MA	AR	ARMA	ARIMA	SARIMA
RMSE	224.67	127.44	121.68	116.79	129.41

Traditional Cloud Approach



Surveillance Scene

CCTV

Cloud

Processing & Monitoring

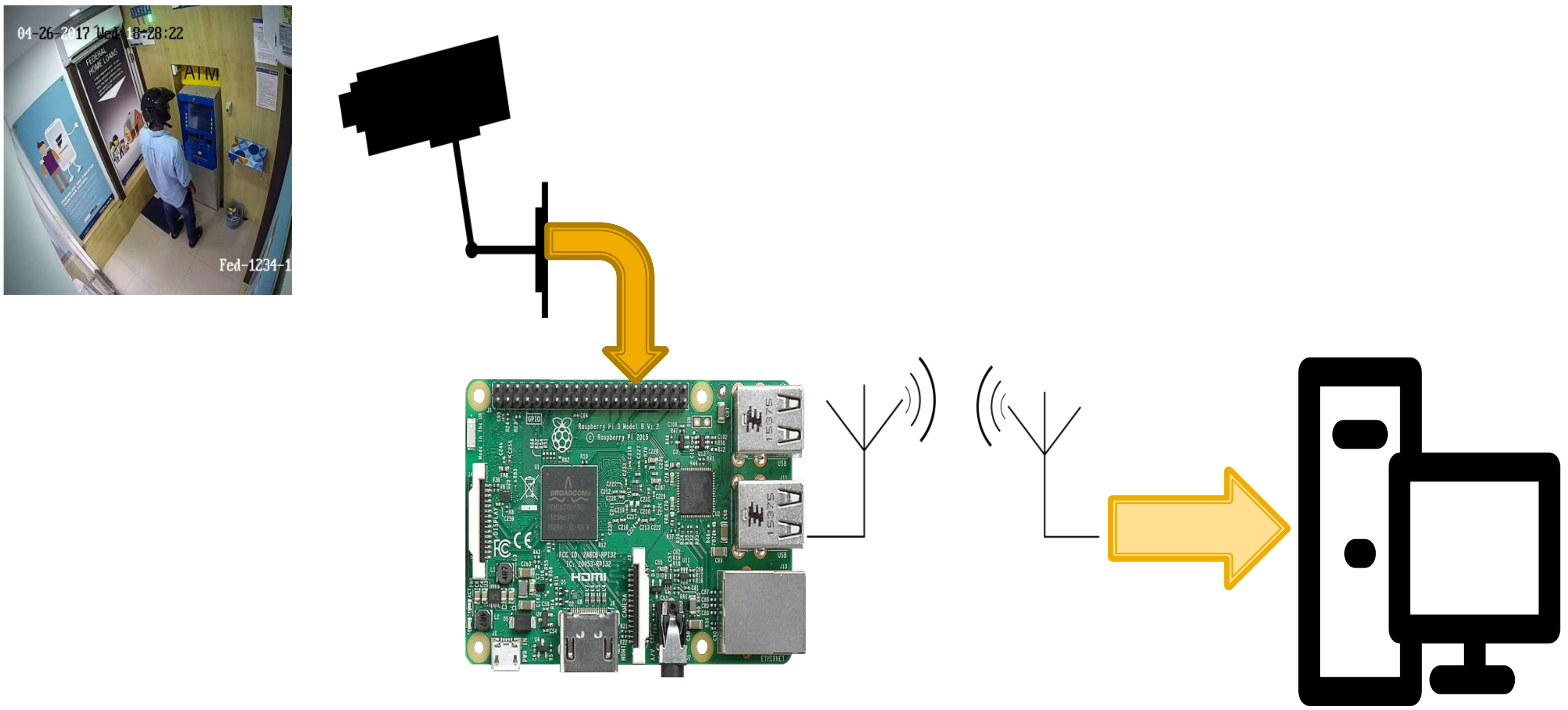
Motivation

- Surveillance for security, patient care
- Impossible for human operator
- Accuracy/subjective

Normal

Abnormal

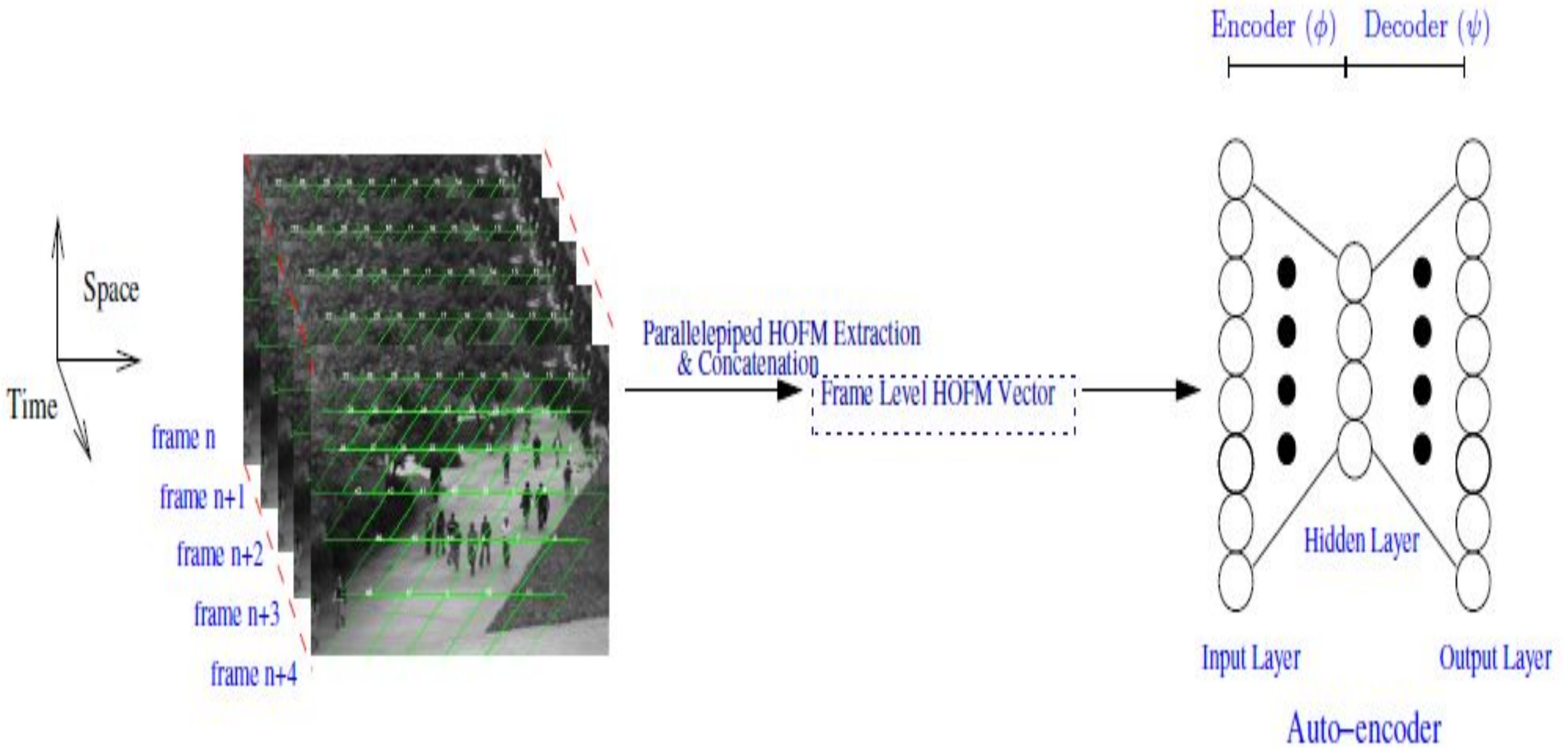
Edge Based Approach



Edge
Processing

Monitoring

Approach



Space
Time
frame n
frame n+1
frame n+2
frame n+3
frame n+4
Parallelepiped HOFM Extraction
& Concatenation
Frame Level HOFM Vector
Encoder (ϕ)
Decoder (ψ)
Hidden Layer
Input Layer
Output Layer
Auto–encoder

Algorithm Abnormality detection using autoencoder

```
procedure ABNORMAL AUTOENCODER(P,ϕ,ψ,X_t))
    P is the set of concatenated HOFM vector of training set.
    ϕ and ψ is the encoder and decoder trained on P.
    X_t is the concatenated HOFM vector of test frames.
    if ||X_t − ψ(ϕ(X_t))||^2 > τ then
        the test frames contain abnormality
    else
        the test frames do not contain abnormality
```

Results

Dataset	AUC	EER (%)	Frame Processing Time (ms)
UCSD Ped1	0.78	29.49	34.85
UCSD Ped2	0.91	15.78	66.47
Subway Entrance	0.84	22.68	15.26
Subway Exit	0.86	19.58	32.85

Results: Time series data - LV panic0 dataset

Results: Time series data - LV panic1 dataset

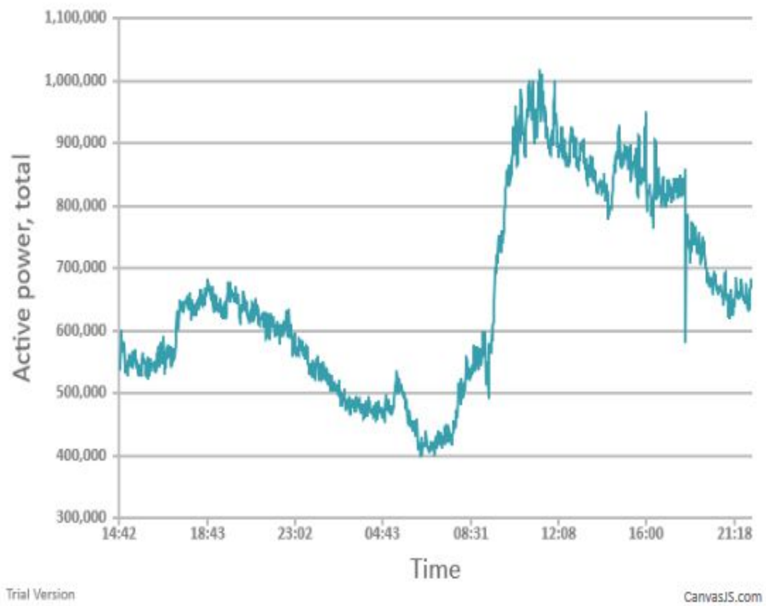
Results: Time series data - LV panic2 dataset

Eg:Real Time Electricity Usage at IITP

24 Hour Data | Data by Date

Show All entries | Search:

Date (dd-mm-yyyy)	Time (hh:mm:ss)	Readings (In standard unit)
06-11-2017	21:59:06	676170.38
06-11-2017	21:58:06	682617.75
06-11-2017	21:57:06	672862.81
06-11-2017	21:56:06	669819.19
06-11-2017	21:55:06	666511.94
06-11-2017	21:54:06	633467.06
06-11-2017	21:53:06	655203.06
06-11-2017	21:52:06	649298.5
06-11-2017	21:51:06	643586.75
06-11-2017	21:50:06	629989.38
06-11-2017	21:49:06	640710.94



Eg:Real Time Electric Energy Usage at IITP

24 Hour Data **Data by Date**

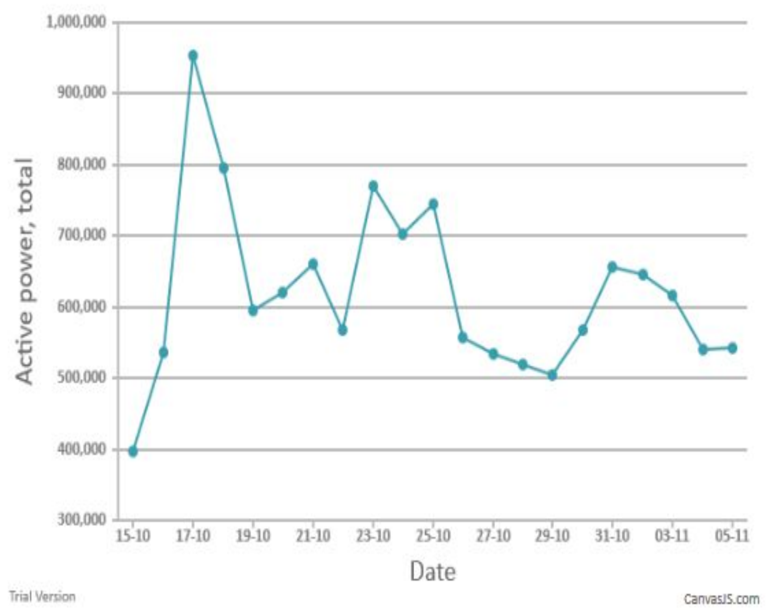
Filter by date 15-10-2017 to 06-11-2017 Proceed

Show All entries

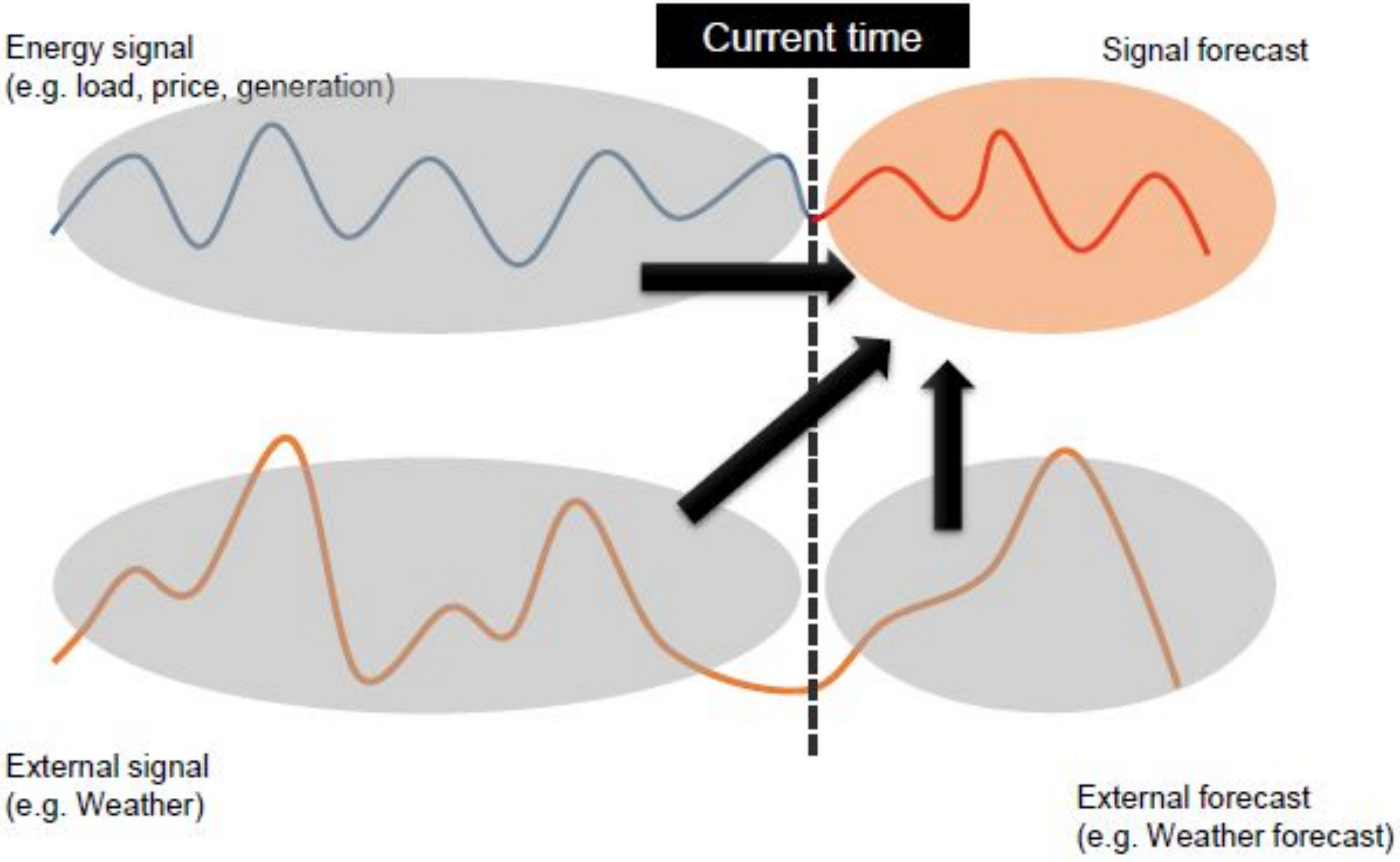
Search:

Date	Average Reading
05-11-2017	541868.95
04-11-2017	540414.56
03-11-2017	616917.99
02-11-2017	645896.68
31-10-2017	656820.58
30-10-2017	567431.32
29-10-2017	504964.03
28-10-2017	519225.13
27-10-2017	534465.36
26-10-2017	556953.28
25-10-2017	744588.47

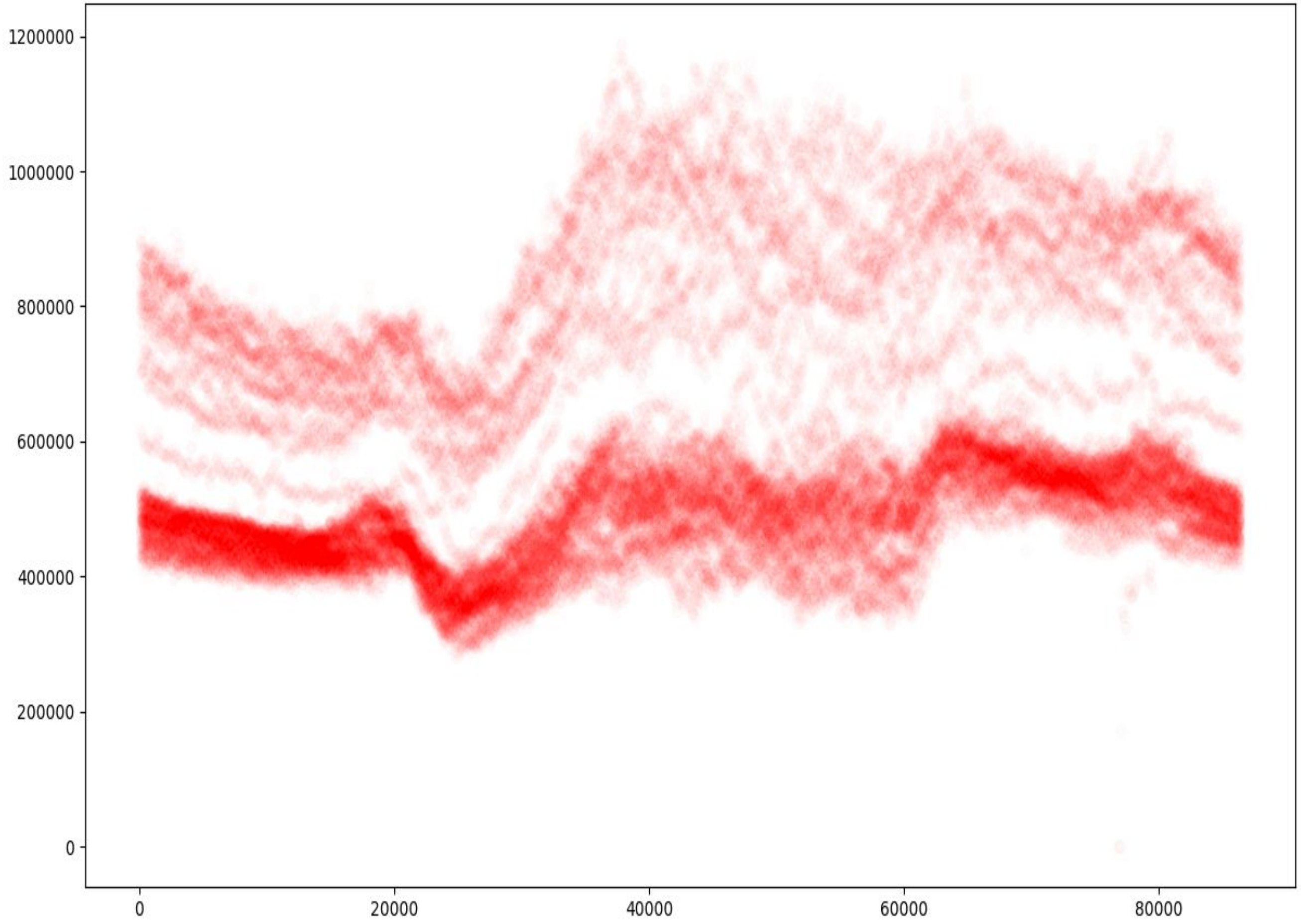
Showing 1 to 21 of 21 entries Previous 1 Next



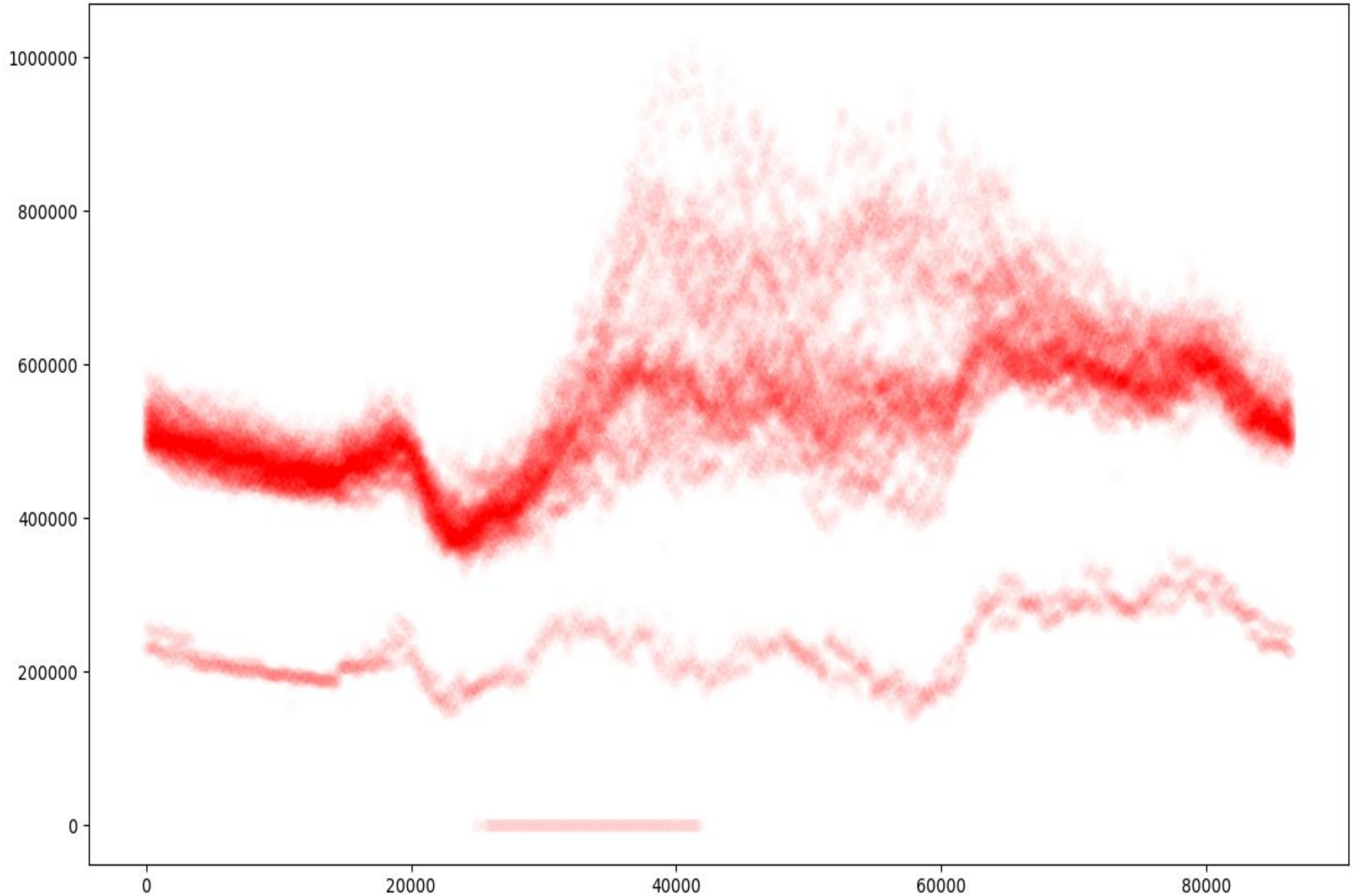
Energy Forecast



Eg:Real Time Electric Energy Usage at IITP



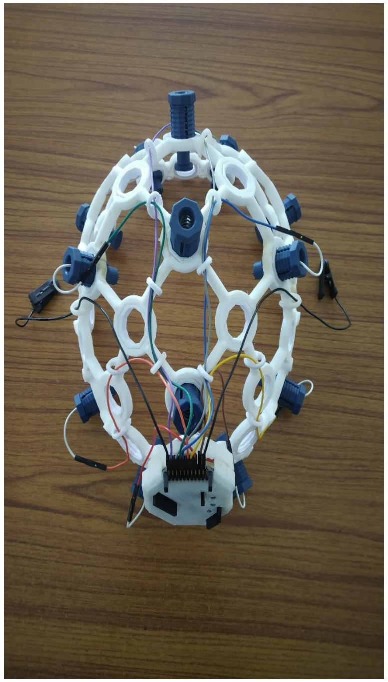
Eg:Real Time Electric Energy Usage at IITP

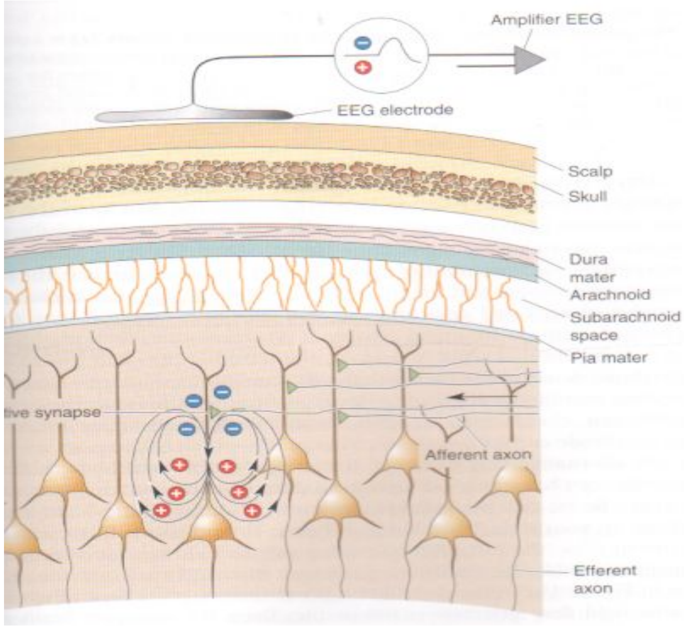


EEG Classification using LSTM and Autoencoder

Electroencephalogram

- The electroencephalogram (EEG) is a recording of the electrical activity of the brain from the scalp.
- During EEG test, small electrodes like cup placed on scalp

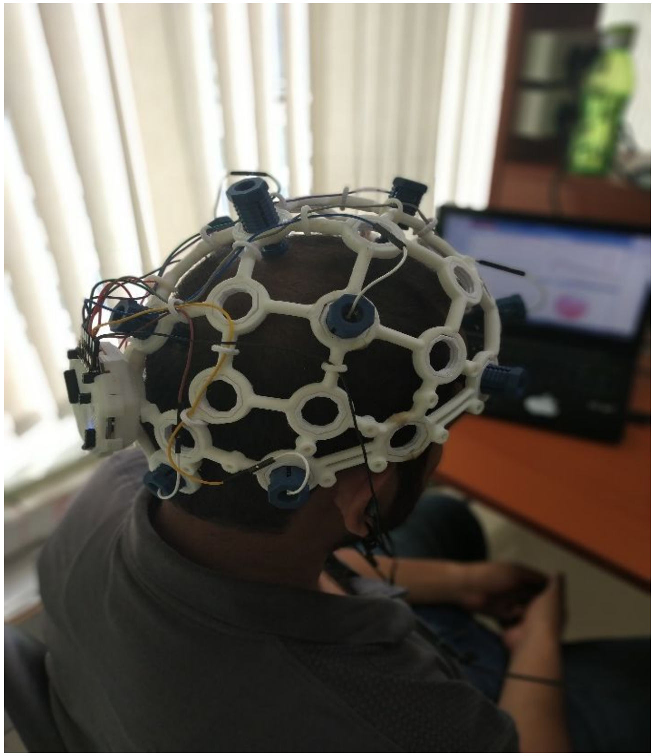


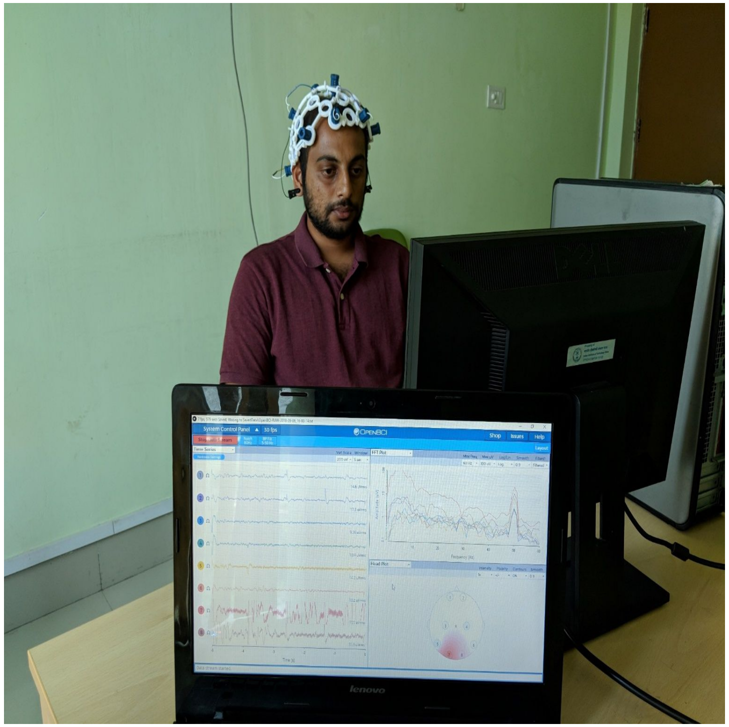


Why EEG ?

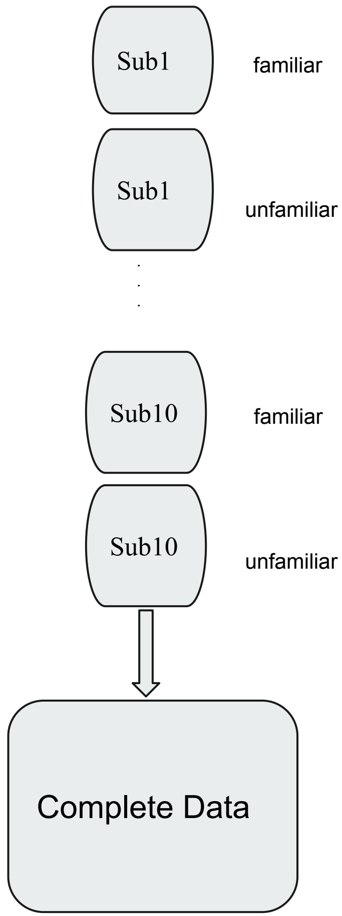
- Always there will be a underlying relation between the lie and the brain signals – difficult to control the reactions inside the brain

- Ability to differentiate known and unknown





Processing Data

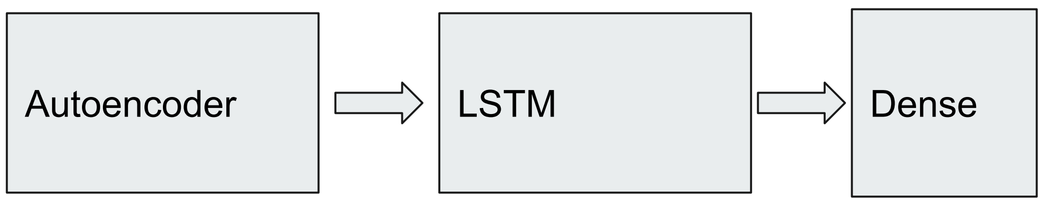


Sample Data set :

P7	P8	O1	O2	case
0.019579	0.031368	0.049292	0.031376	1
-0.005074	0.061788	0.172507	-0.009579	1
-0.252178	0.067108	0.328600	-0.344678	1
-0.619706	0.456606	0.759965	-0.801646	1
-0.627086	1.868949	1.796608	-0.784960	1

Deep Learning Models

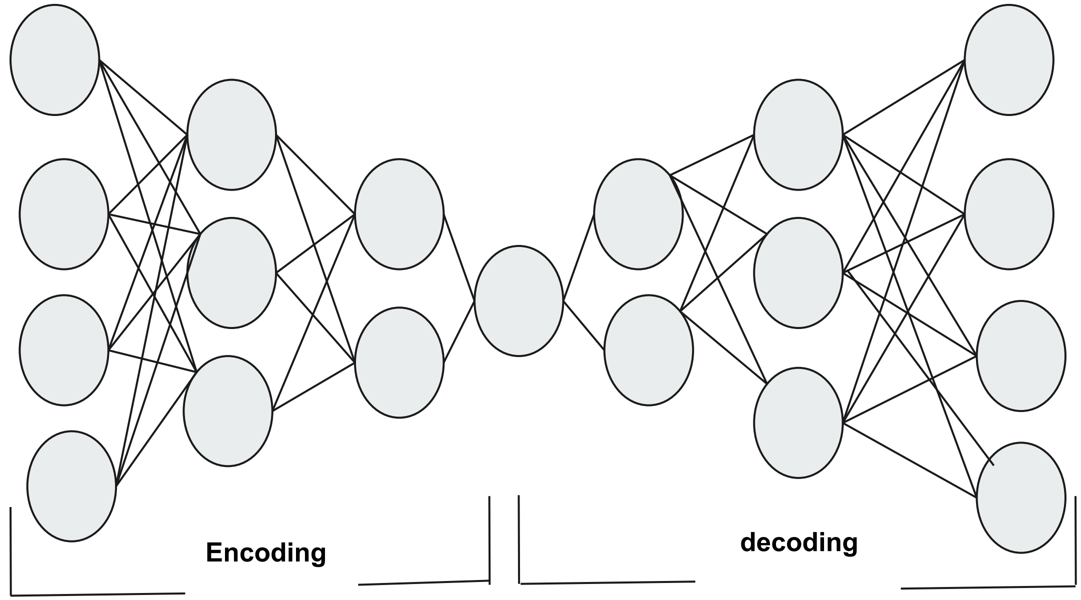
Ensembling of AUTOENCODER and LSTM:



LSTM Architecture

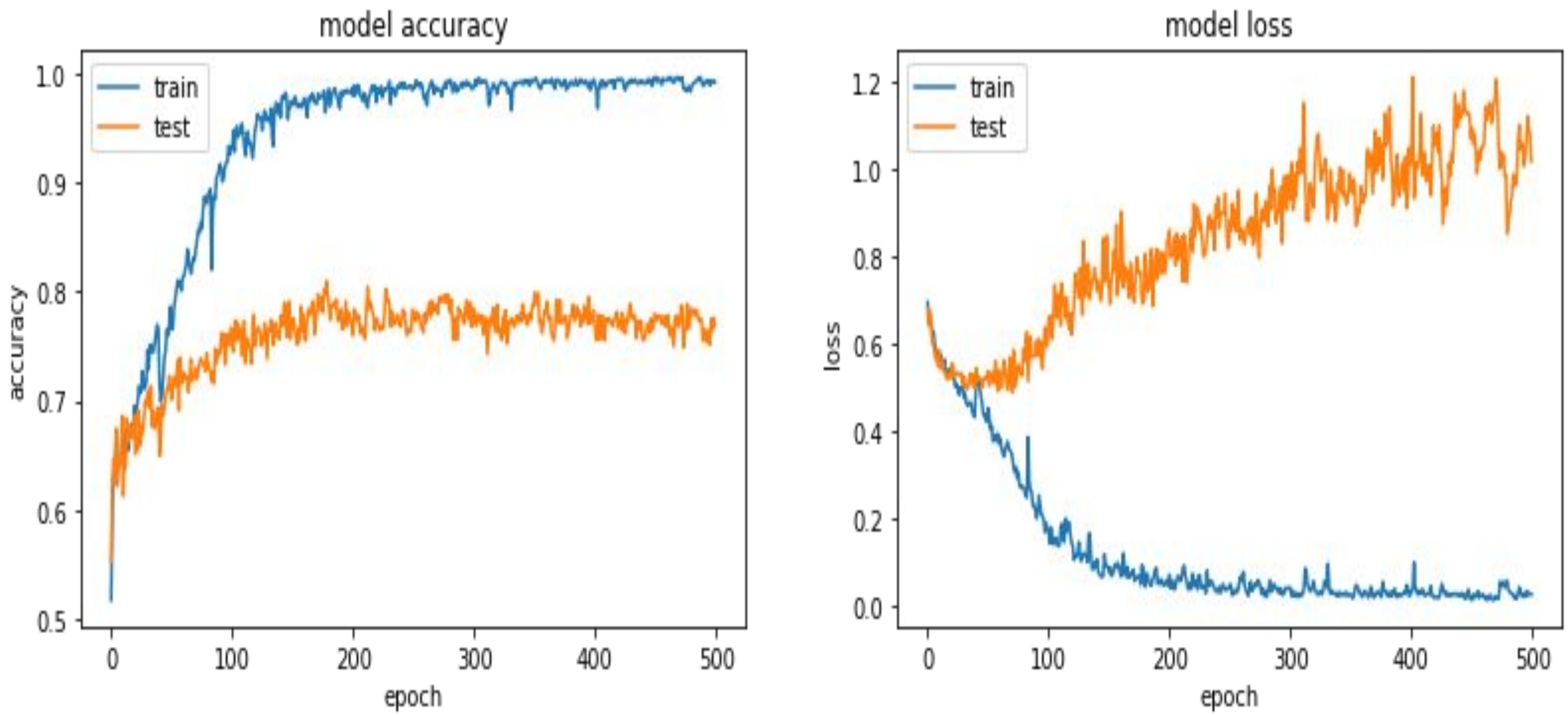
Layer (type)	Output Shape	Param #
dropout_3 (Dropout)	(None, 128, 4)	0
lstm_2 (LSTM)	(None, 128)	68096
dropout_4 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 128)	16512

Autoencoder Architecture

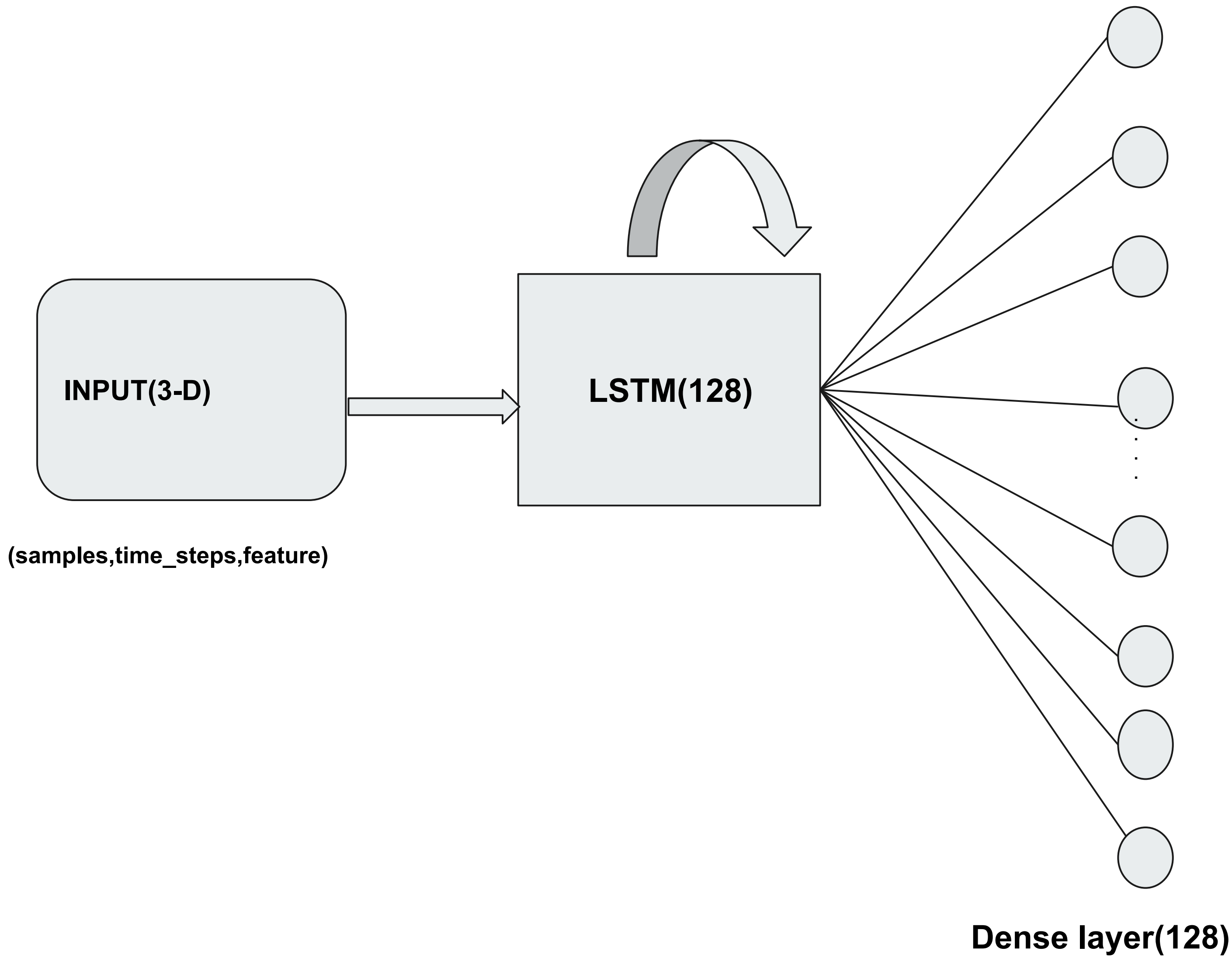


- Input is feed to autoencoder
- Output of autoencoder is image of input
- Reshape the output in 3D as(number of sample, time steps, features)
- Feed to LSTM

Results


model accuracy
train
test
accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0
100
200
300
400
500
epoch
model loss
train
test
loss
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
100
200
300
400
500
epoch

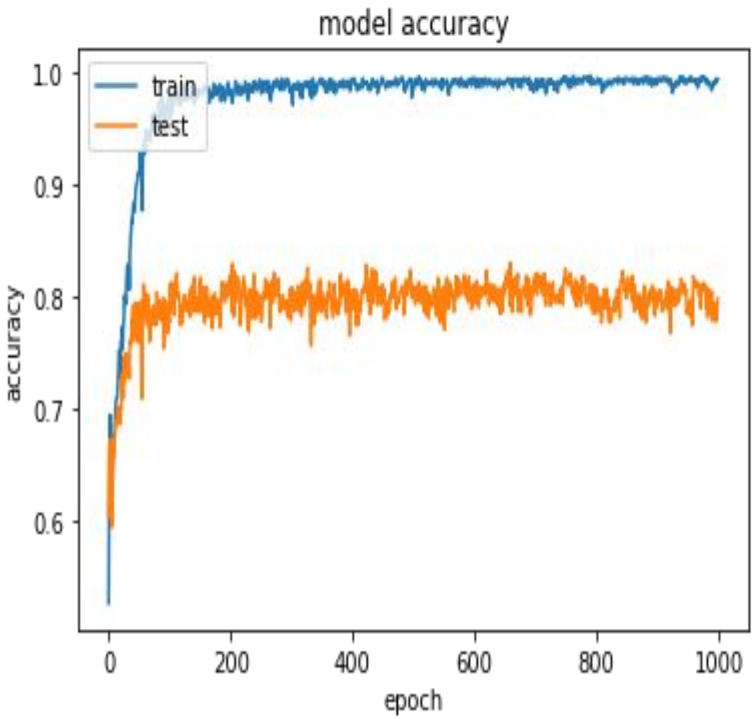
Simple LSTM Model:

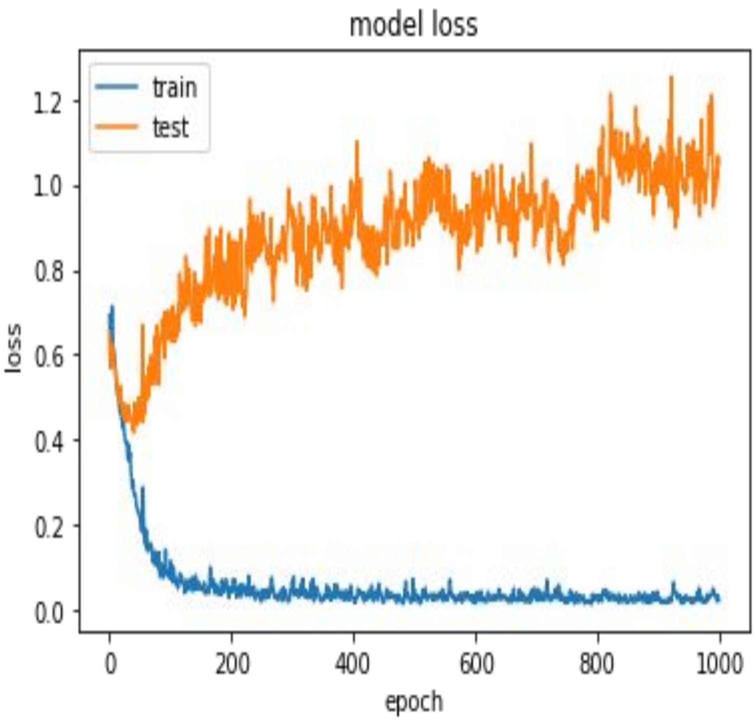


Network Architecture

Layer (type)	Output Shape	Param #
dropout_3 (Dropout)	(None, 128, 4)	0
lstm_2 (LSTM)	(None, 128)	68096
dropout_4 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 128)	16512

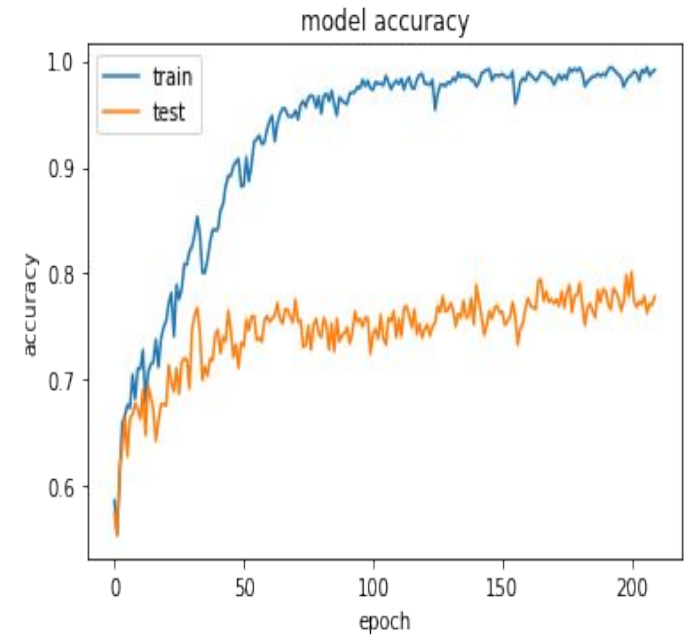
Results

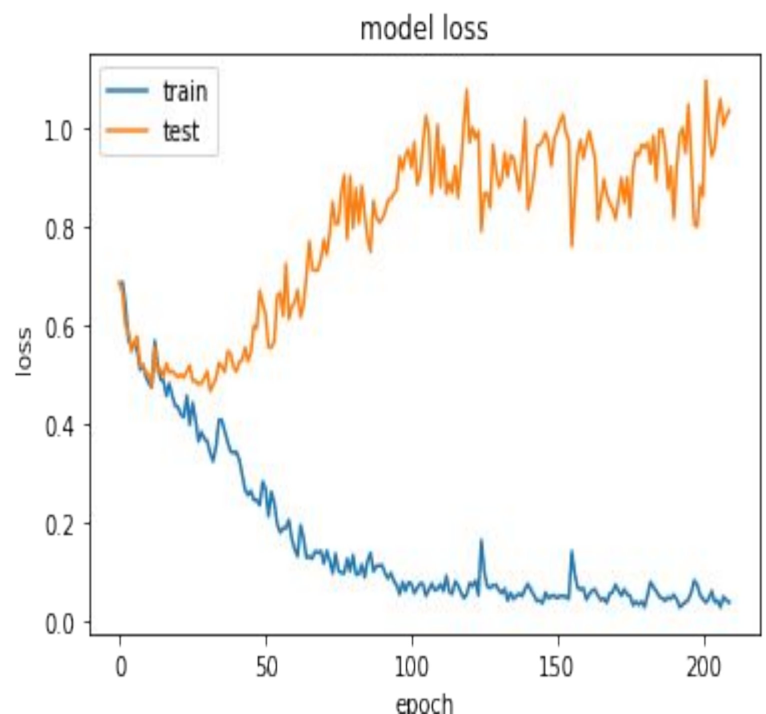




Stacked LSTM:

Results





Network Architecture

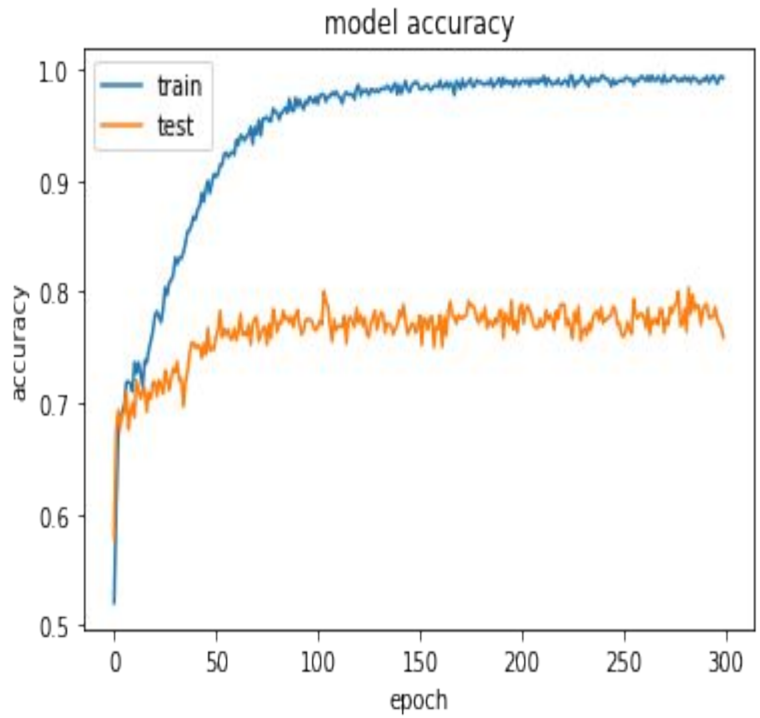
Layer (type)	Output Shape	Param #
dropout_13 (Dropout)	(None, 128, 4)	0
lstm_9 (LSTM)	(None, 128, 128)	68096
dropout_14 (Dropout)	(None, 128, 128)	0
lstm_10 (LSTM)	(None, 128)	131584
dropout_15 (Dropout)	(None, 128)	0
dense_5 (Dense)	(None, 128)	16512

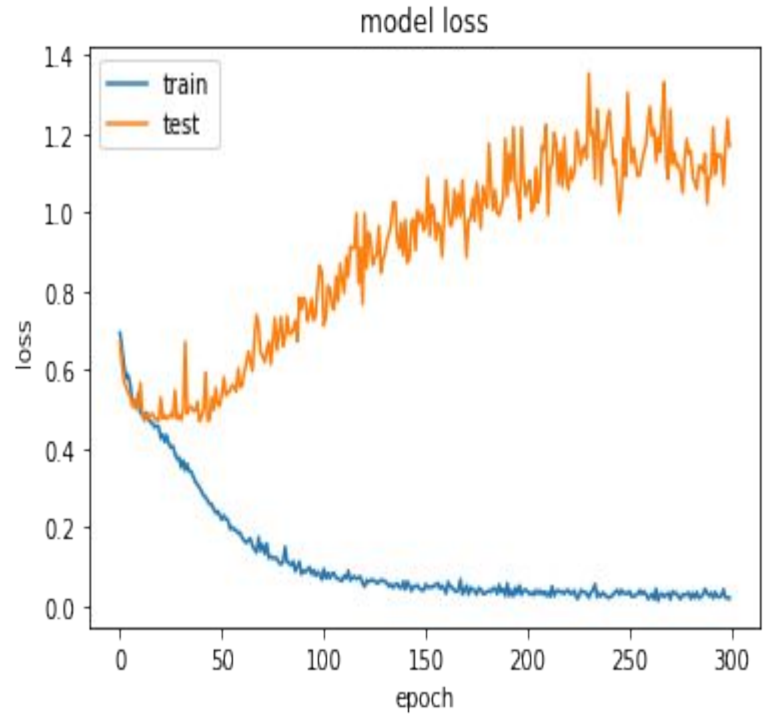
Stateful LSTM:

Network Architecture

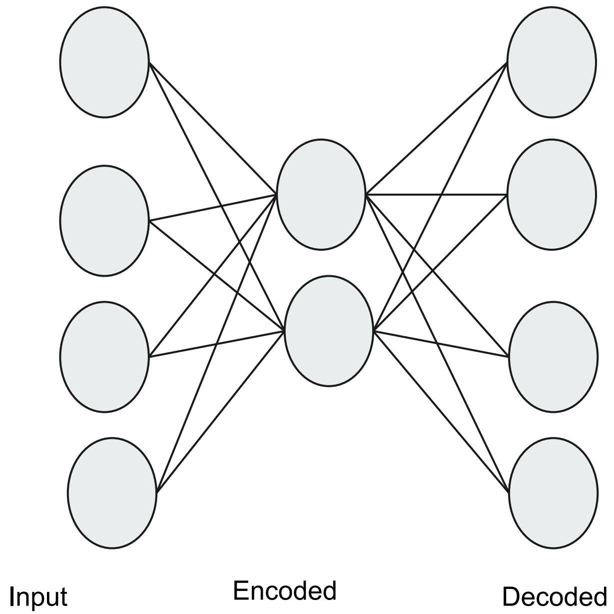
Layer (type)	Output Shape	Param #
lstm_1 (LSTM)	(1, 128)	68096
dropout_1 (Dropout)	(1, 128)	0
dense_1 (Dense)	(1, 128)	16512

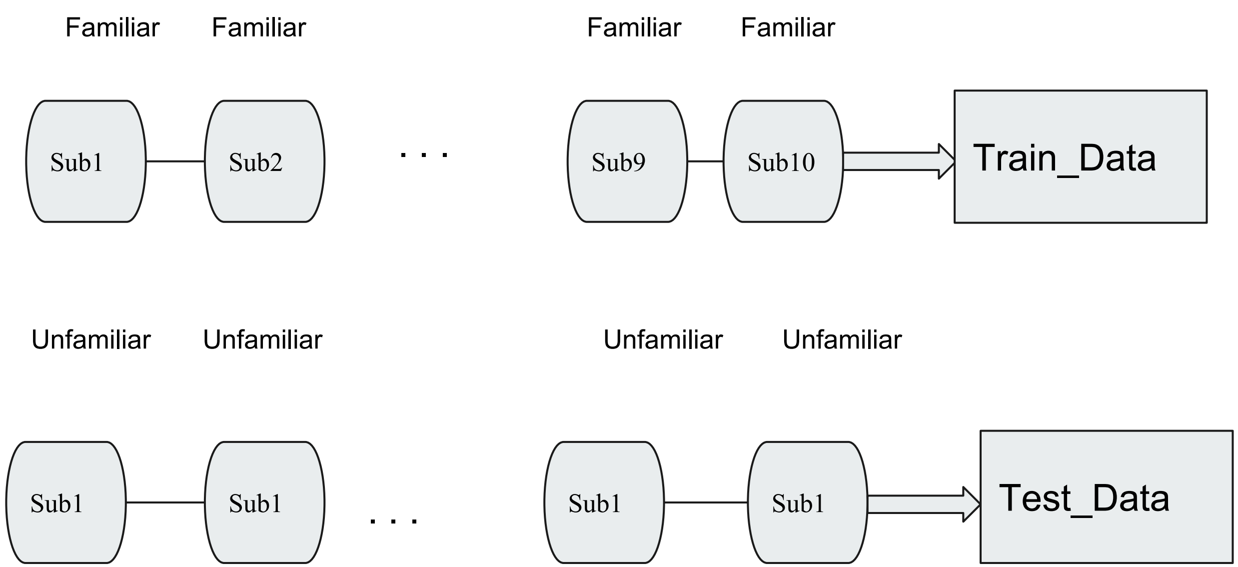
Results


model accuracy
train
test
accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0
50
100
150
200
250
300
epoch

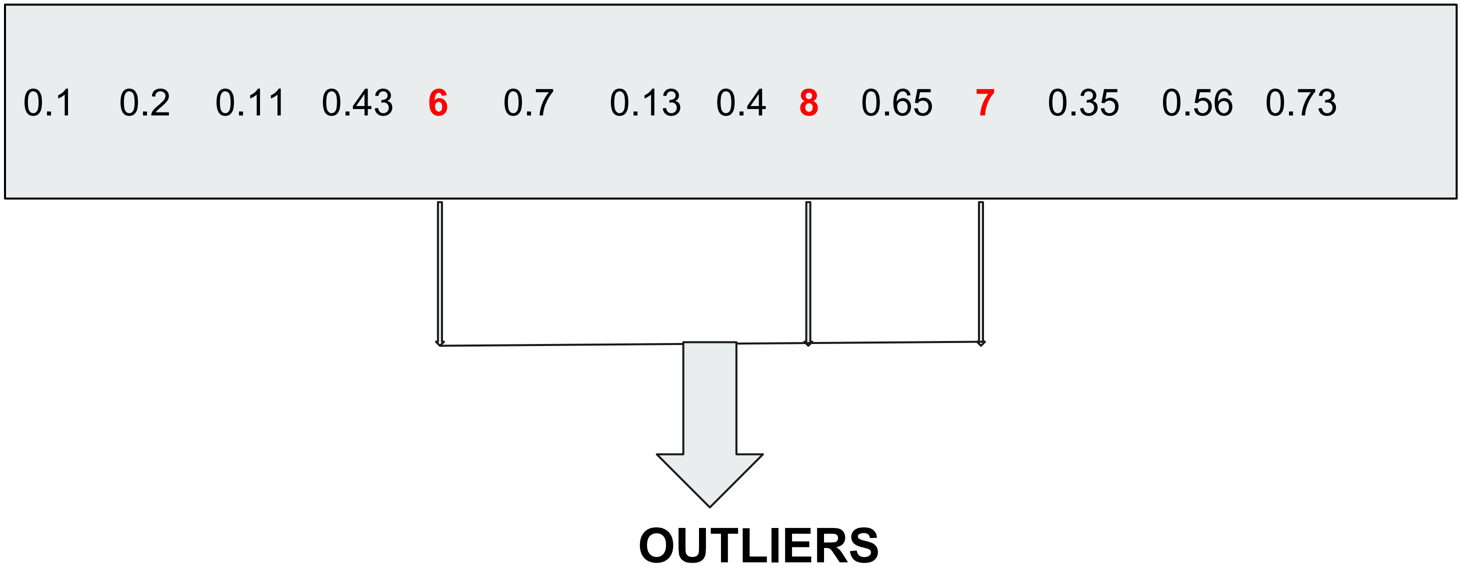

model loss
train
test
loss
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
200
250
300
epoch

Simple Autoencoder:





- We are using the concept of outlier detection



- We can find mean and standard deviation
- Calculate threshold based on mean and standard deviation
- Points above threshold are **OUTLIERS**

Reconstruction Error

0.456 0.321 0.157 0.056 4.562 0.112 0.156 0.442 6.32 0.152 5.63 0.211 0.143 0.11

Mean = 1.3448571428571428

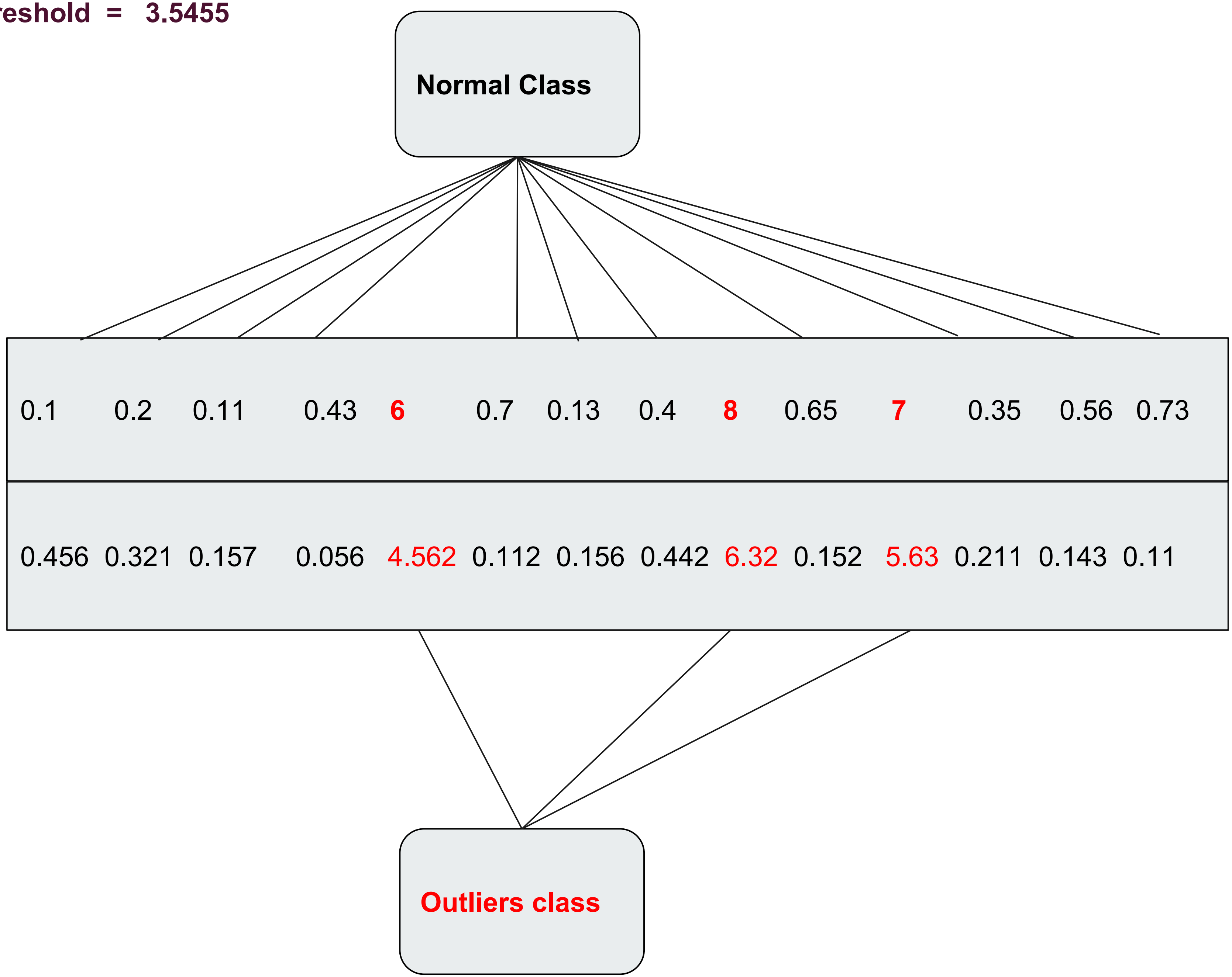
STD = 2.2006712241347395

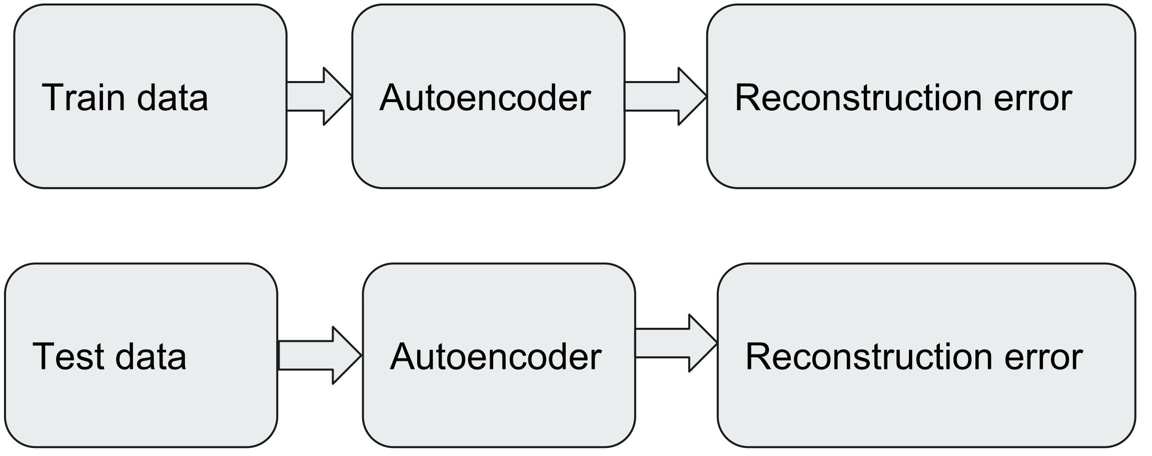
Threshold = 3.5455283669918822

- Data points which are above threshold are **OUTLIERS**

- We can use this concept for classification

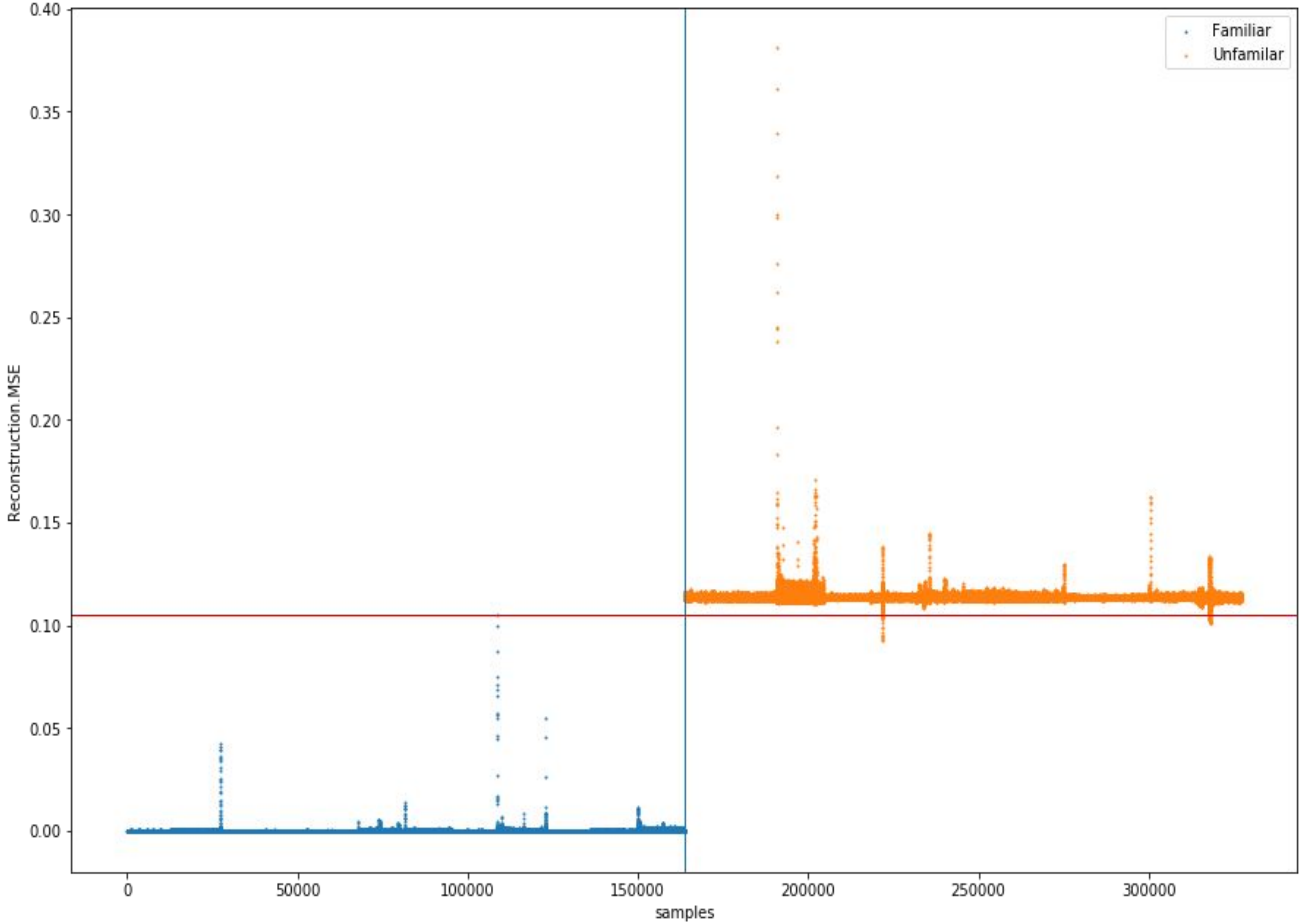
Threshold = 3.5455



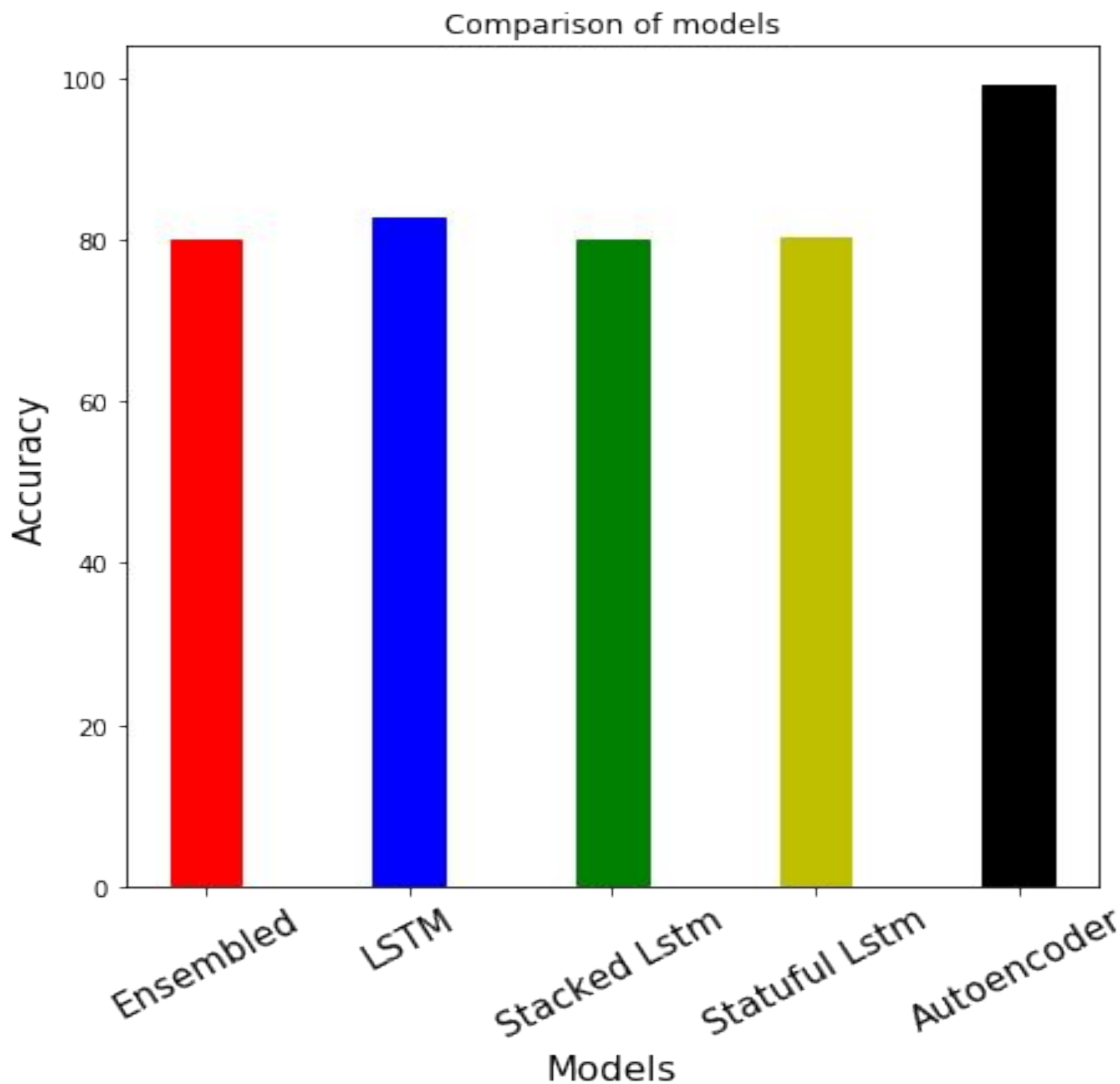


- RMSE for reconstruction error
- Calculate threshold of reconstruction error
- Classify test data which is unfamiliar data based on threshold.

Result


Familiar
Unfamilar
Reconstruction.MSE
samples
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
50000
100000
150000
200000
250000
300000

Comparison of models



References

[1] V. abootalebi, m. h. moradi, and m. a. khalilzadeh, a new approach for eeg feature ex- traction in p300-based lie detection, computer meth- ods and programs in biomedicine, vol. 94, no. 1 2009.

[2] Z. h. e. tan, k. g. smitha, and a. p. vinod, detection of familiar and unfamiliar images using eeg-based brain-computer interface, in systems, man, and cybernetics (smc), 2015 ieee international con- ference on. ieee, 2015.

[3] https://github.com/cerebro409/eeg-classification-using-recurrent-neural-network.

[4] https://www.kaggle.com/imrandude/h2o-autoencoders-and-anomaly-detection-Python.

[5] M. teplan et al., fundamentals of eeg measurement, measurement science review, vol. 2, no. 2, pp. 111, 2002.

[6] F. lotte, m. congedo, a. lcuyer, f. lamarche, and b. arnaldi, a review of classification algorithms for eeg-based braincomputerinterfaces, journal of neural engineering, vol. 4, no. 2, p. r1, 2007.

Thanks

Questions