NC STATE UNIVERSITY

CSC 591: Machine Learning for User Adaptive Systems

EEG-Based Activity Recognition with ML

Kartik Rawool (khrawool)
Aaron Dias Barreto (aadiasba)
Vikram Pande (vspande)

Introduction

- Electroencephalograms (EEGs) can record electrical activity in the brain.
- They can be used to augment human sensory functions or control robotic devices.
- In order to perform these functions the Brain Computer Interface (BCI) must be able to classify EEG patterns as corresponding to a certain task and relay that information to control the device of interest.

Introduction

- We focus on the BCI competition III Dataset V in which the goal is to classify three mental tasks online.
- There are 3 tasks:
  - Imagination of repetitive self-paced left hand movements, (left, class 2),
  - Imagination of repetitive self-paced right hand movements, (right, class 3),
  - Generation of words beginning with the same random letter, (word, class 7).

BCI Competition III (dataset V): http://www.bbci.de/competition/iii/

Dataset

- BCI Competition iii Dataset V
- 32 Electrodes to collect EEG data.
- **EEG**: ElectroEncephaloGraphy - method to record an electrogram of the spontaneous electrical activity of the brain.
- Sampling rate is 512 Hz

BCI Competition III (dataset V): http://www.bbci.de/competition/iii/

Dataset

- 3 Subjects with 3 activities over 4 sessions
- Dataset has raw EEG signals, and precomputed features by spatial filtering and calculating power spectral density.
- **PSD**: the measure of signal's power content versus frequency.

BCI Competition III (dataset V): http://www.bbci.de/competition/iii/

Dataset

- **Size**
  - Each subject has 3 labeled training files and 1 unlabeled testing file.
  - 31216 Training samples.
  - 10464 Testing samples.
  - Raw EEG signals for 32 channels and 96-dimensional precomputed features.

BCI Competition III (dataset V): http://www.bbci.de/competition/iii/

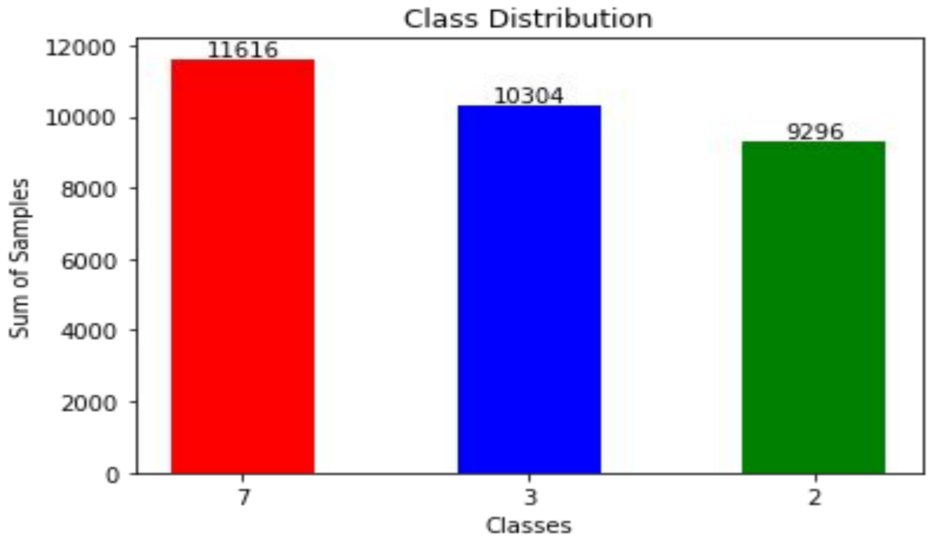
Methodology

1. Preprocessing the data & Exploratory Data Analysis
2. Training with 5 models
3. Evaluation
4. Comparison

Methodology: Preprocessing and EDA

1. Total 9 Train files: 3 for 3 subjects.
2. Total 3 Test files
3. Used:
    a. 7 as **Train** out of 9
    b. 2 as **Val** out of 9
    c. 3 as **Test**

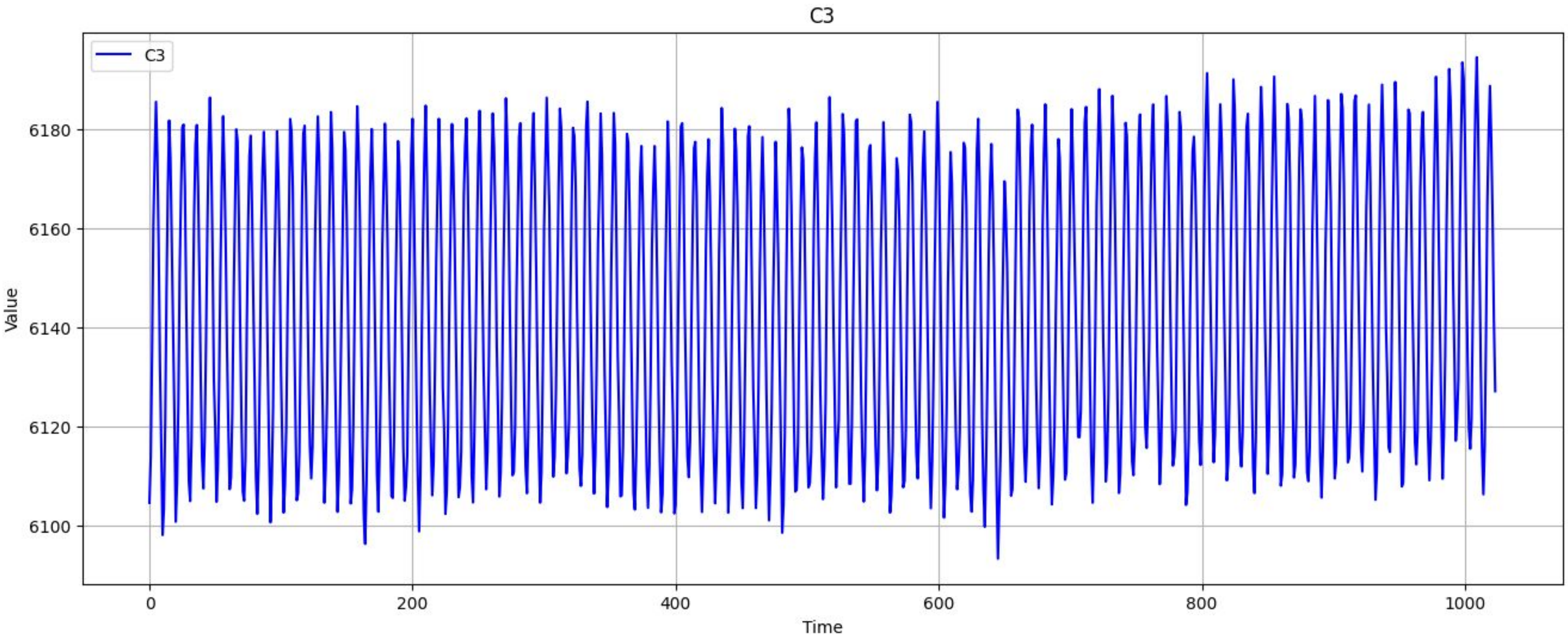
Dataset: EDA:Training data distribution



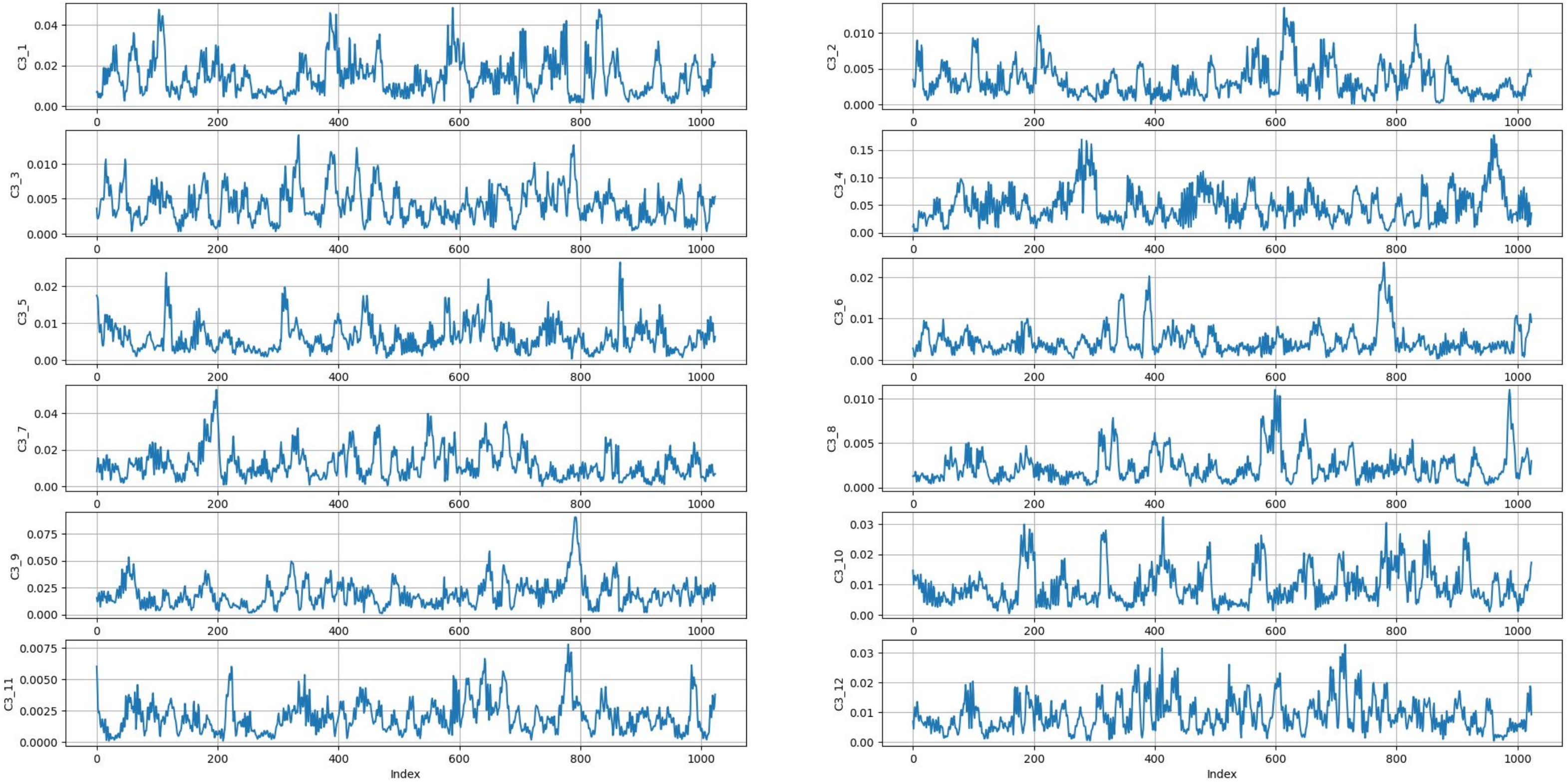

```
m={2:0,3:1,7:2} #Mapping of classes
{0: 1.1346889690608943, 1: 1.044427672955975, 2: 0.8611491391827422} # Balanced Weight Assignment
```

BCI Competition III (dataset V): http://www.bbci.de/competition/iii/

Raw Signal - C3



After PSD - C3



Methodology: Modeling

1. SVM
2. kNN
3. Hidden Markov Model
4. LSTM
5. BiLSTM

Methodology: Modeling: SVM

1. Supervised Learning
2. Implemented GridSearchCV

```
▾ GridSearchCV
GridSearchCV(estimator=SVC(),
             param_grid={'C': [0.1, 1, 10, 100], 'gamma': [1, 0.1, 0.01, 0.001],
                         'kernel': ['rbf']},
             verbose=10)
▾ estimator: SVC
SVC()
▾ SVC
SVC()
```

Selected Parameters:

'C': 100,

'gamma': 1,

'kernel': 'rbf'

Methodology: Modeling: kNN

1. Supervised Learning
2. Implemented GridSearchCV

```
GridSearchCV
GridSearchCV(estimator=KNeighborsClassifier(),
             param_grid={'n_neighbors': [50, 100, 200],
                         'weights': ['uniform', 'distance']},
             verbose=10)
estimator: KNeighborsClassifier
KNeighborsClassifier()
KNeighborsClassifier
KNeighborsClassifier()
```

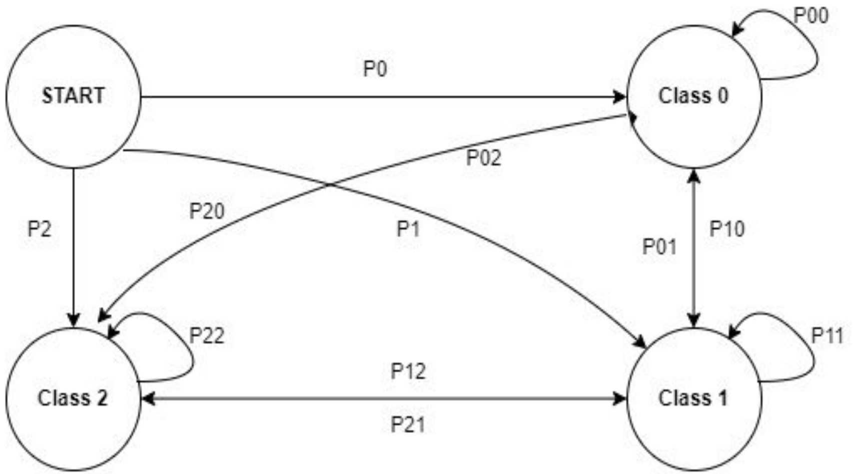
Selected parameters:

n_neighbors = 100

weights = uniform

Methodology: Modeling: Hidden Markov Model

1. Probabilistic approach
2. States = Classes = 3
3. Transition from one class to another.
4. Gaussian Mixture Model (GMM) for each class
5. Train GMMHMM using the GMMs for each class



HMM flow with classes as States

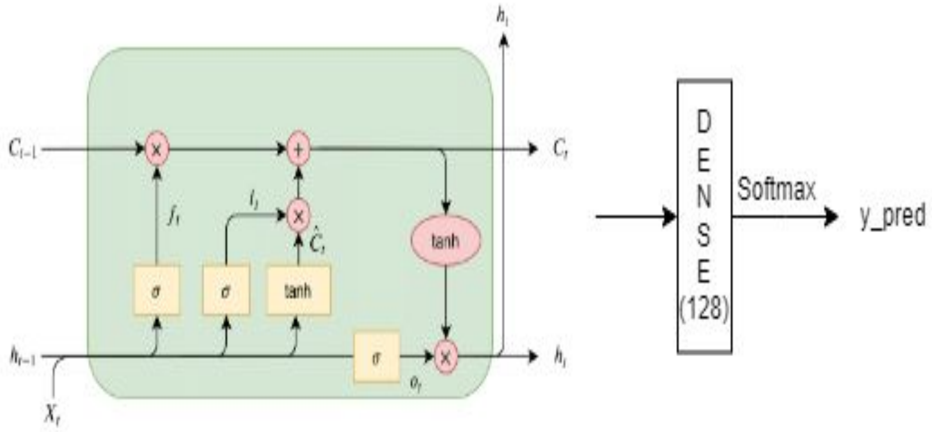
Methodology: Modeling: Long-Short-Term Memory

1. Type of Recurrent Neural Network (RNN)
2. Designed to handle sequential data such as Time Series
3. Capable of learning long-term dependencies.
4. Contains
    a. Input gates: Allow in optional information from current cell state
    b. Forget gates: Control flow of information
    c. Output gates: Update and finalize the next hidden state

Methodology: Modeling: Long-Short-Term Memory

Hyperparameters

1. Window size: 16
2. Input shape: 96
3. Optimizer: Adam
4. Learning Rate: 0.01
5. Activation: ReLU
6. Dropout: 0.7
7. Epochs: 20
8. Batch size: 32

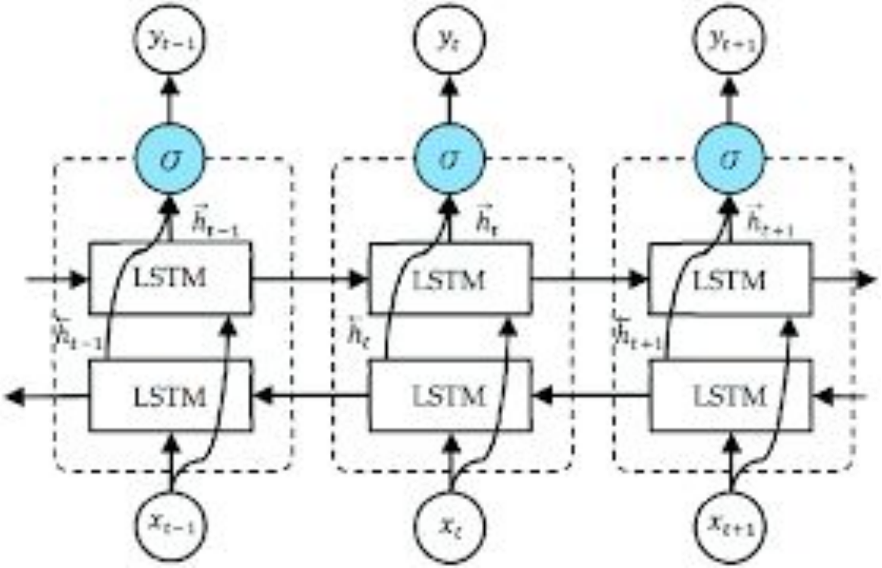


Methodology: Modeling: BiDirectional Long-Short-Term Memory

1. Advancement of LSTM.
2. Uses two LSTM layers.
3. One layer processes input in the forward direction.
4. Can learn bidirectional long-term dependencies between time steps of time-series or sequence data
5. Backtracking easier

Methodology: Modeling: BiDirectional Long-Short-Term Memory

1. Window size: 16
2. Input shape: 96
3. Optimizer: Adam
4. Learning Rate: 0.001
5. Activation: ReLU
6. Dropout: 0.5
7. Epochs: 20
8. Batch Size: 32



Methodology

Attempted LSTM with Attention Layer

- not much improvement in accuracy.
- more training time.

Attempted wavelet and feature selection

- Not a significant difference

Improvements:

- 1D CNN + Attention + LSTM
- 1D CNN to capture temporal information, attention to capture important temporal information, LSTM for long-term dependencies

Methodology: Evaluation

Metrics used for Evaluation:

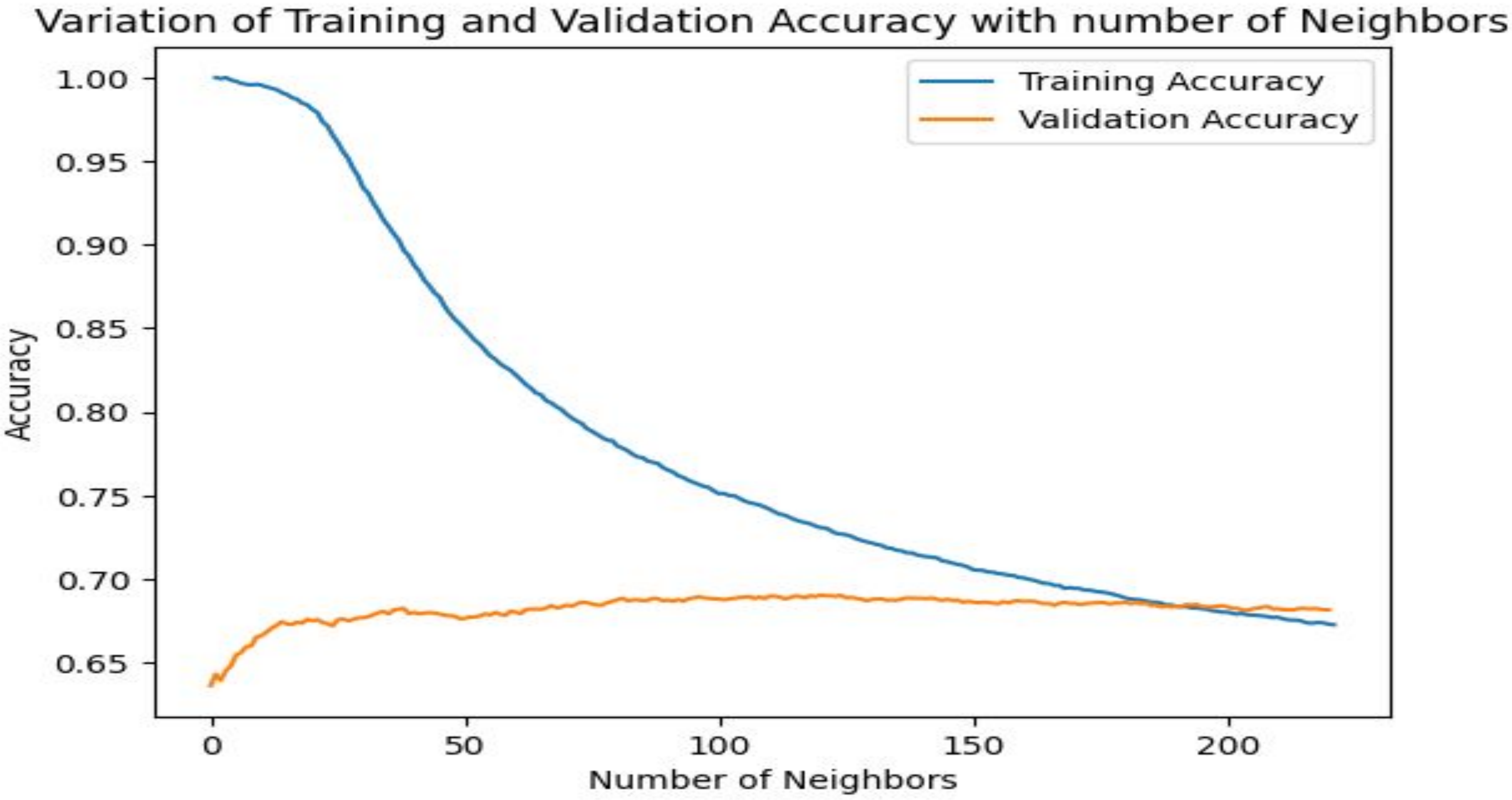
a. Accuracy
b. F1 measure
c. Precision
d. Recall

Results

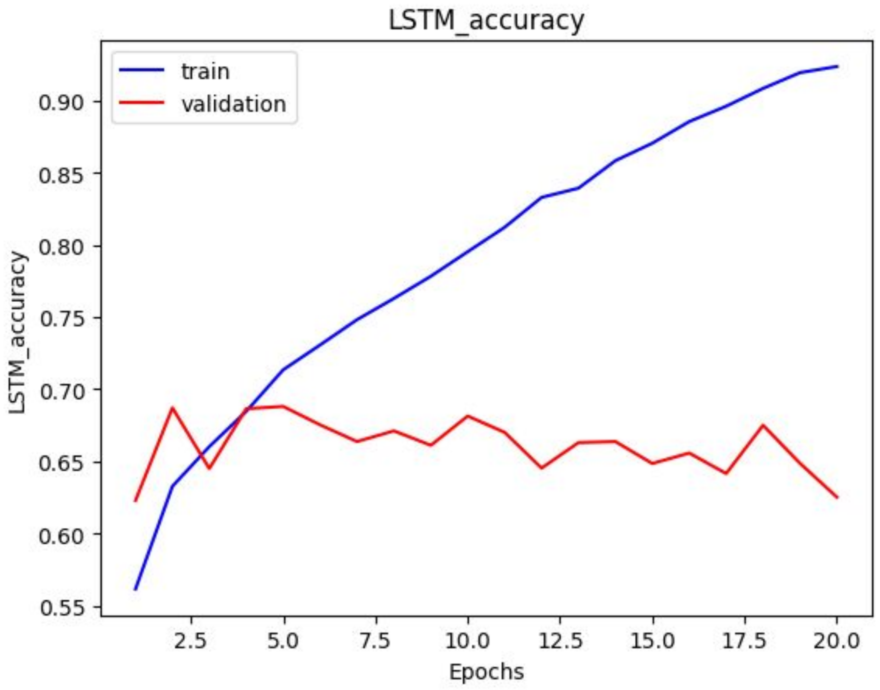
Models	Training				Validation			
	Accuracy	F1	Precision	Recall	Accuracy	F1	Precision	Recall
SVM	0.67	0.67	0.68	0.67	0.65	0.64	0.65	0.64
kNN	0.75	0.75	0.75	0.76	**0.69**	**0.68**	0.68	0.68
HMM	0.38	0.36	0.40	0.44	0.46	0.46	0.48	0.47
LSTM	0.92	0.92	0.92	0.91	0.62	0.62	0.62	0.62
BiLSTM	0.96	0.96	0.96	0.96	**0.66**	**0.66**	0.66	0.65

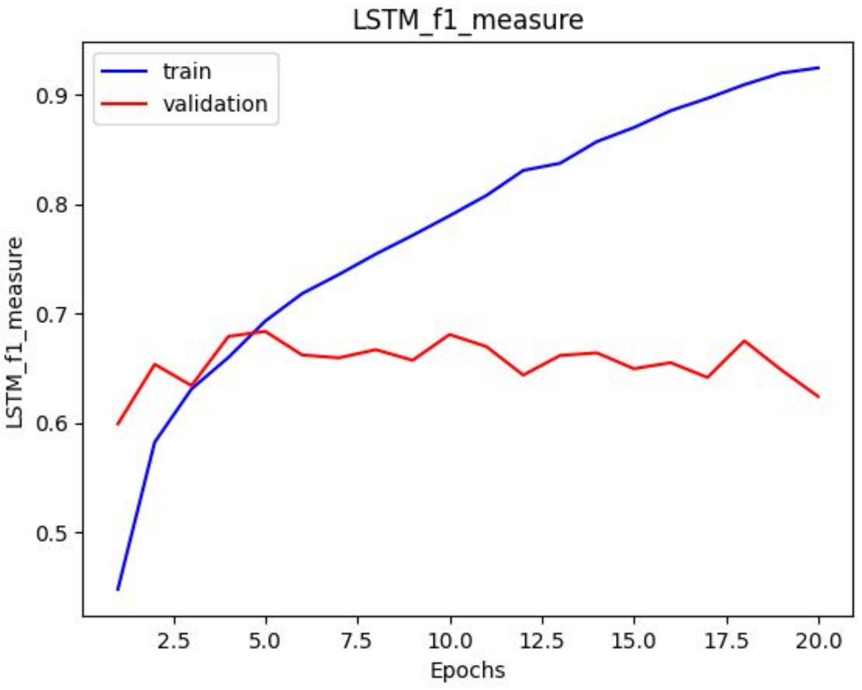
Predictions on Test dataset (unknown labels).

kNN Accuracy

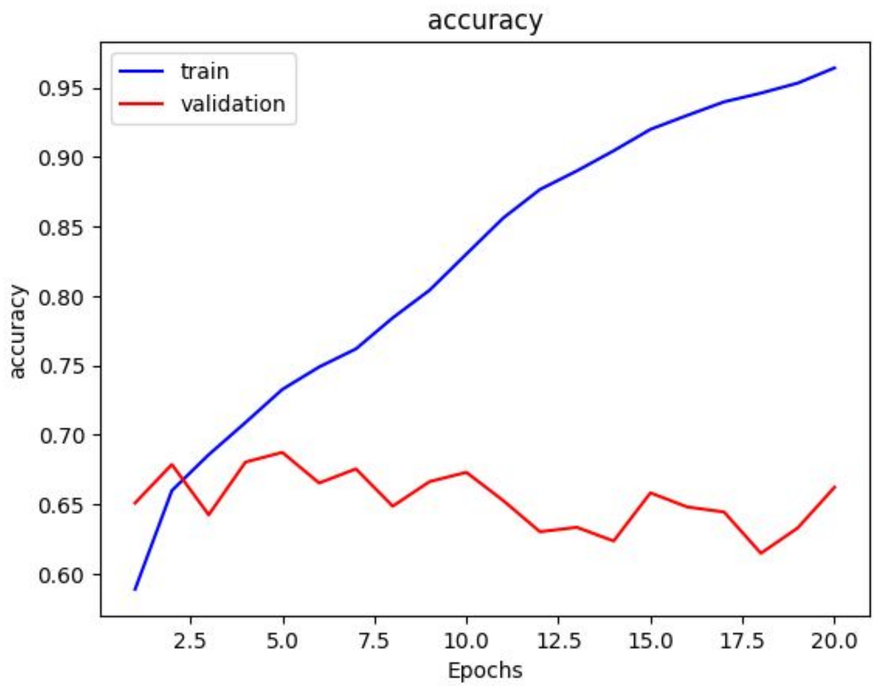


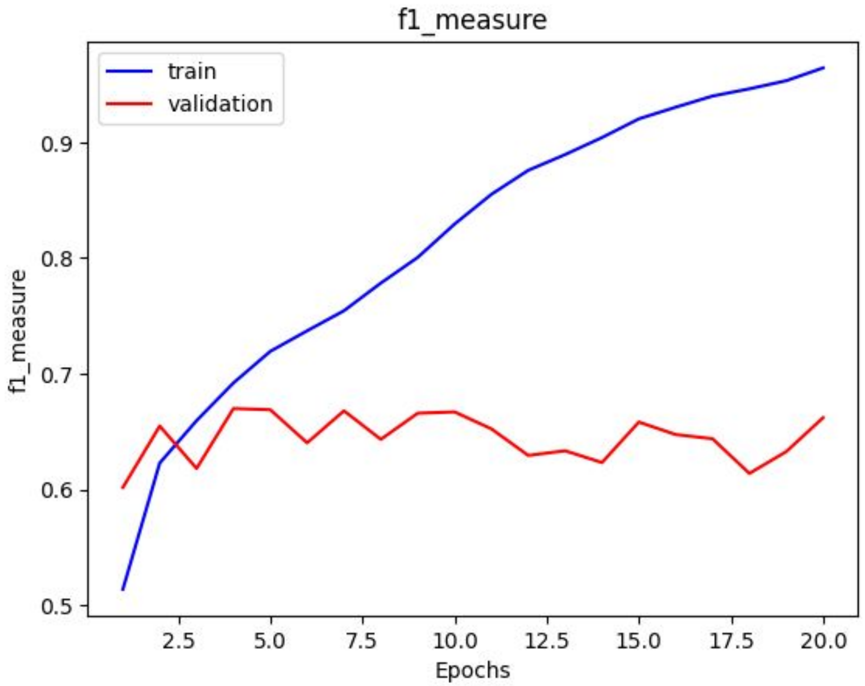
Results: Accuracy & F1 - LSTM





Results: Accuracy & F1 - BiLSTM





Competition Results

#.	contributor	psd	acc	s1	s2	s3	research lab	co-contributors
1.	Ferran Galan	y	**68.65**	79.60	70.31	56.02	University of Barcelona	Francesc Oliva, Joan Guardia
2.	Xiang Liao	y	**68.50**	78.08	71.66	55.73	University of Electronic Science and Technology of China (UESTC)	Yu Yin, Dezhong Yao
3.	Walter	y	**65.90**	77.85	66.36	53.44	???	
4.	Xiaomei Pei	y	**65.67**	76.03	69.36	51.61	Institute of Biomedical Engineering of Xi'an Jiaotong University	Guangyu Bin, Chongxun Zheng
5.	Irene Sturm	y	**64.91**	78.08	63.83	52.75	Fraunhofer FIRST (IDA), Berlin	Guido Dornhege
6.	Stephan Uray	y	**64.60**	81.05	73.04	39.68	TU Graz	
7.	Julien Kronegg	y	**64.04**	76.06	64.83	51.18	University of Geneva	Douglas Rofes
8.	John Q. Gan	y	**63.91**	77.40	63.83	50.46	University of Essex, Colchester	Louis C.S. Tsui
9.	Shiliang Sun	n	**62.83**	74.31	62.32	51.99	Tsinghua University, Beijing	Changshui Zhang, Jie Pan
10.	J. Ignacio Serrano M. D. del Castillo	y	**62.61**	75.80	61.75	50.23	Instituto de Automatica Industrial. CSIC. Madrid	
11.	Changshui Zhang	y	**60.47**	72.15	59.22	50.00	Tsinghua University, Beijing	Shiliang Sun, Feiping Nie
12.	Douglas Rofes	y	**59.81**	72.52	59.85	46.99	University of Geneva	
13.	Alois Schloegl	n	**52.71**	69.00	57.05	32.29	TU Graz	Carmen Vidaurre
14.	Ehsan Arbabi	n	**50.25**	55.41	51.79	43.61	Sharif University of Technology	Mohammad Bagher Shamsollahi
15.	Remy Lehembre	y	**50.23**	72.60	46.31	31.65	Universite Catholique de Louvain-la-Neuve (UCL-Belgium)	Simon Cedric
16.	Georgios Lappas	y	**45.72**	71.78	33.81	31.39	Technological Educational Institution (TEI) of Western Macedonia, University of Hertfordshire	Andreas Albrecht
17.	Mohammad Bagher Shamsollahi	y	**44.97**	71.46	32.52	30.76	Sharif University of Technology	Ehsan Arbabi
18.	Ikaro Silva	y	**30.68**	38.98	27.45	25.55	???	
19.	Ali Salehi	n	**27.97**	26.54	32.84	24.53	???	
20.	Ikaro Silva2	y	**14.24**	5.82	10.54	26.38	???	

www.bbci.de/competition/iii/results/index.html

Challenges

1. Raw data format as collected from EEG signals
2. Time-series classification require more deep analysis and structured models.
3. Validation loss was increasing due to overfitting and dataset complexity (sudden changes).

Conclusion

- Simple Machine Learning Algorithms such as **kNN** outperformed Deep Learning models (RNNs)
  - LSTMs were not able to model the temporal dependence of the data.
  - Data did not have a significant temporal dependence.
- HMM did not perform as well as expected.
  - Markovian Assumption is not valid.
- Subject 3 predictions difficult.

References

[1] Song, Le, Byron Boots, Sajid M. Siddiqi, Geoffrey J. Gordon and Alex Smola. "Hilbert Space Embeddings of Hidden Markov Models." International Conference on Machine Learning (2010).

[2] L. Rabiner and B. Juang, "An introduction to hidden Markov models," in IEEE ASSP Magazine, vol. 3, no. 1, pp. 4-16, Jan 1986, doi: 10.1109/MASSP.1986.1165342.

[3] Clinical Brain-Computer Interface Challenge 2020 (CBCIC at WCCI2020): Overview, methods and results Anirban Chowdhury, Member, IEEE, and Javier Andreu-Perez, Senior Member, IEEE

Thank You