Project Proposal

P300-Based Real-Time EEG Speller with Customizable Machine Learning Models
Mentors:

The objective of this project is to build a real-time Brain–Computer Interface (BCI) speller based on the **P300 Event-Related Potential (ERP)** using **OpenViBE**, while extending it beyond the default classical ML pipeline.
The project guides students across the full workflow of EEG-based BCI systems—from signal acquisition and preprocessing to model training and real-time classification.
The core contribution is integrating advanced machine learning models such as **SVM** and **EEGNet** into OpenViBE's P300 speller framework, enabling improved accuracy and customizable classification modules.

The final outcome is a fully functional P300 speller system capable of detecting user-intended characters through EEG responses, with a flexible backend that supports multiple ML models.

Work to be Done:

1. Foundations of EEG and P300 Speller Paradigm

■ Understand EEG signals, ERP components, and the physiological basis of the P300 response.
■ Study the row–column flashing paradigm used in classical P300 spellers.
■ Explore OpenViBE's P300 speller demo scenario and its default processing pipeline.

---

2. EEG Signal Acquisition & Preprocessing

■ Learn event tagging, epoch extraction, and stimulus alignment.
■ Apply standard preprocessing methods: band-pass filtering, artifact handling, baseline correction, and downsampling.
■ Visualize ERP responses and confirm the presence of P300 peaks around 300 ms.

---

3. Feature Engineering & Baseline Classifier

■ Extract features from EEG epochs (time-domain samples, PCA/CSP features).
■ Implement baseline classifiers such as LDA or Logistic Regression.
■ Evaluate baseline P300 vs non-P300 classification accuracy.

---

4. Integrating Classical Machine Learning Models

■ Implement SVM, Random Forest, and other supervised ML models in Python.
■ Train models on extracted ERP features and compare performance against baseline.
■ Export trained ML models (pickle, ONNX, etc.) for real-time usage.

---

5. Deep Learning for EEG: EEGNet

■ Study the EEGNet architecture designed for ERP classification.
■ Implement EEGNet using Keras/PyTorch with input shaped for P300 epochs.
■ Train, validate, and optimize deep models for enhanced P300 detection.
■ Compare deep learning vs classical ML in terms of accuracy, latency, and robustness.

---

6. OpenViBE Integration of Custom Models

■ Use Python Scripting Box or external TCP/VRPN communication to connect models with OpenViBE.
■ Build a custom classification module that loads trained SVM/EEGNet models.
■ Feed real-time EEG epochs from OpenViBE to the model and send predictions back.
■ Ensure seamless integration into the P300 speller decision-making loop.

---

7. Real-Time P300 Speller System

■ Combine preprocessing, classifier, and spelling interface into a unified real-time pipeline.
■ Implement confidence scoring, majority voting, and trial averaging to improve reliability.
■ Test system with multiple users and assess spelling accuracy and selection speed.
■ Analyze error patterns and optimize flashing parameters, epoch length, and classifier thresholds.

---

8. Final Deliverables

- A fully functional P300-based EEG speller system in OpenViBE.
- Modular ML backend allowing plug-and-play support for SVM, EEGNet, or other models.
- Performance evaluation across different ML models (accuracy, ITR, latency).
- Documentation, OpenViBE scenarios, trained model files, and user guide.
