# VIDEO SURVEILLANCE AND ERP-BASED BCIS AS ANOMALY DETECTION: NEW METHODS AND DATASET

by
MEHMET YAĞAN

Submitted to the Graduate School of Engineering and Sciences
in partial fulfilment of
the requirements for the degree of Master of Sciences

Sabancı University
July 2022

# ABSTRACT

VIDEO SURVEILLANCE AND ERP-BASED BCIS AS ANOMALY DETECTION: NEW METHODS AND DATASET

MEHMET YAĞAN

Electronics Engineering M.Sc. THESIS, JULY 2022

Thesis Supervisor: Dr. Hüseyin Özkan

Keywords: Anomaly Detection, Recurrent Neural Networks, P300 Speller, Future Feature Prediction, Multiple Instance Learning

Security cameras are widely used to detect and prevent crimes, but the number of surveillance videos has increased due to this prevalence. By processing these videos with the help of a suitable machine learning algorithm, unfavorable events can be brought to the attention of experts to manually monitor. Since these unfavorable events are of various types and few in number, this problem can be addressed in the anomaly detection framework. In this thesis, several new anomaly detection algorithms have been developed using the UCF-Crime dataset. First of all, features were extracted from the videos in the dataset with the help of a pre-trained artificial neural network (ANN). Then, the size of these features was reduced with a semisupervised ANN, an autoencoder (AE) or principal component analysis (PCA). Lastly, anomaly detection was performed using an recurrent neural network (RNN) based on future feature estimation by regression. In addition to these algorithms, a large scale anomaly detection dataset has been introduced. Due to their non-invasive nature, one of the most commonly used event related potentials in brain–computer interface (BCI) system designs is the P300 electroencephalography (EEG) signal. In order to train and test P300-based BCI speller systems in more realistic high speed settings, there is an arising requirement for a large and challenging benchmark dataset. Various datasets already exist in the literature but most of them are not publicly available, and they either have a restrictive number of subjects or utilize relatively long stimulus duration (SD) and inter-stimulus intervals (ISI). The use of long ISI, in particular, not only reduces the speed and the information

transfer rates (ITRs) but also simplifies the P300 detection. This leaves a limited challenge to the state-of-the-art machine learning and signal processing algorithms. Therefore, one certainly needs a large-scale dataset in challenging settings to fully exploit the recent advancements in algorithm design (machine learning and signal processing) and achieve high-performance speller results. To this end, by using 32-channels EEG, here we introduce a new freely and publicly accessible P300 dataset, hoping to enhance research findings towards building efficient BCIs. The introduced dataset is composed of 18 subjects performing a 40-target ($5 \times 8$) cued-spelling task, with reduced SD (66.6 ms) and ISI (33.3 ms) for fast spelling. We have also processed, analyzed, and character-classified the introduced dataset and presented the accuracy and ITR results as a benchmark.

# ÖZET

ANOMALİ TESPİTİ OLARAK VİDEO GÖZETİMİ VE OLAY İLGİLİ POTANSİYEL TABANLI BBA: YENİ YÖNTEMLER VE VERİ SETİ

MEHEMT YAĞAN

Elektronik Mühendisliği YÜKSEK LİSANS TEZİ, TEMMUZ 2022

Tez Danışmanı: Dr. Hüseyin Özkan

Anahtar Kelimeler: Anomali Tespiti, Özyinelemeli Sinir Ağı, P300 Heceleme sistemi, Gelecek Öznitelk Tahmini, Çoklu Öğe Öğrenmesi

Güvenlik kameraları suçları tespit etmek ve önlemek amacıyla yaygın olarak kullanılmaktadır, ancak bu yaygınlık nedeniyle gözetleme videolarının sayısı artmıştır. Bu videolar uygun bir makine öğrenme algoritması yardımıyla işlenerek, istenmeyen olaylar izlenmesi için uzmanların dikkatine sunulabilir. Bu olumsuz olaylar çeşitli tiplerde ve az sayıda olduğundan, bu sorun anomali tespiti ile giderilebilir. Bu tezde, UCF-Crime veri seti kullanılarak birkaç yeni anomali tespit algoritması geliştirilmiştir. Öncelikle veri kümesindeki videolardan, önceden eğitilmiş bir yapay sinir ağı (YSA) yardımıyla öznitelikler çıkarılmıştır. Daha sonra bu özniteliklerin boyutu yarı denetimli bir YSA, bir özkodlayıcı veya temel bileşenler analizi ile küçültüldü. Son olarak, regresyon yoluyla gelecek öznitelik tahminine dayalı olarak bir özyinelemeli sinir ağı kullanılarak anomali tespiti yapıldı. Bu algoritmalara ek olarak, büyük ölçekli bir anomali tespit veri seti de yayınlanmıştır. İnvaziv olmayan yapıları nedeniyle, beyin-bilgisayar arayüzü (BBA) sistem tasarımlarında en yaygın olarak kullanılan olayla ilgili potansiyellerden biri P300 elektroensefalografiyi (EEG) sinyalidir. P300 tabanlı BBA heceleme sistemlerini daha gerçekçi yüksek hız ayarlarında eğitmek ve test etmek için, büyük ve zorlu bir veri setine ihtiyaç duyulmaktadır. Literatürde halihazırda çeşitli veri kümeleri mevcuttur, ancak bunların çoğu halka açık değildir, sınırlı sayıda denek içerirler ya da nispeten uzun uyaran süresi (US) ve uyaranlar arası aralıklar (UAA) kullanırlar. Özellikle uzun UAA kullanımı, yalnızca hızı ve bilgi aktarım hızlarını (BAH) azaltmakla kalmaz, aynı zamanda P300 algılamasını da basitleştirir. Bu, son teknoloji makine öğrenimi ve

sinyal işleme algoritmaları için sınırlı bir zorluk bırakır. Bu nedenle, algoritma tasarımındaki son gelişmelerden tam olarak yararlanmak ve yüksek performanslı heceleme sonuçları elde etmek için, zorlayıcı ve büyük ölçekli bir veri kümesine ihtiyaç vardır. Bu amaçla, 32 kanallı EEG kullanarak, verimli BBA'lar oluşturmaya yönelik araştırma bulgularını geliştirmeyi umarak, ücretsiz ve halka açık yeni bir P300 veri seti sunuyoruz. Sunulan veri seti, hızlı yazım için düşük US (66.6 ms) ve UAA (33.3 ms) ile 40 hedefli (5×8) bir yazım görevi gerçekleştiren 18 denekten oluşur. Ayrıca tanıtılan veri setini işlenmiş, analiz edilmiş ve doğruluk ve BAH sonuçları bir kıyaslama olarak sunulmuştur.

## ACKNOWLEDGEMENTS

I would like to thank my advisor Assist. Prof. Dr. Hüseyin Özkan for his guidance. I also would like to thank Assoc. Prof. Şuayb Arslan and Assist. Prof. Dr. Nihan Alp for guiding me through my endeavors.

I am grateful to my committee members, Assist. Prof. Dr. Sinan Yıldırım and Assist. Prof. Dr. Tuna Çakar for their insightful comments and questions.

This thesis study was supported by The Scientific and Technological Research Council of Turkey (TUBITAK) under Contract 118E268.

*Dedicated to my family*

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

# 1. INTRODUCTION

Anomalies are events occurring in the real world that are not expected (Chandola, Banerjee & Kumar, 2016). It can range from road accidents to network attacks and health problems (Kim, Hong & Park, 2021; Ten, Hong & Liu, 2011). Since it has such a huge scope, there have been numerous studies in this area. Two main requirements to develop efficient anomaly detection methods are large datasets containing normal and anomalous data and machine learning methods that can learn to separate these normal and anomalous occurrences reliably. In this thesis we propose new anomaly detection algorithms and produce a new, large, open access dataset.

For the algorithm proposal part of the thesis, we choose anomaly detection in surveillance videos because of the increasing number of surveillance cameras and need for automated anomaly detection systems. We developed supervised and unsupervised anomaly detection methods using UCF-Crime dataset (Sultani, Chen & Shah, 2018).

It is hard to produce natural anomaly detection datasets due to anomalies being rare occurrences. Because of this most datasets either simulate anomaly videos using actors (Boiman & Irani, 2005) or have limited number of anomalies (Mahadevan, LI, Bhalodia & Vasconcelos, 2010). As one of the main contributions of this thesis, we introduced a new dataset consisting of natural anomalies that are easy to produce and annotate. For this purpose, we used oddball paradigms to create anomalous EEG brain signals. Oddball paradigm presents an unexpected stimulus during a stream of expected stimuli, as a response brain produces P300 event related potential (ERP) waves (Duncan-Johnson & Donchin, 1977). While we can control timing and presentation of the oddball events, brain response to the unexpected event is natural. We measured and recorded brain signals via an electroencephalogram (EEG) device.

In this chapter, a summary of our video anomaly detection methods is provided in Section 1.1, a summary of the proposed dataset is provided in Section 1.2 and the organization of the thesis is given in Section 1.3.

## 1.1 Video Anomaly Detection

Security cameras all over the world have an important role at identification of criminal or life threatening activities and providing quick responses. As a result of increasing number of surveillance cameras are increasing continuously, due to increasing number of security cameras, overseeing all the footage became a costly task that requires considerable amount of human power. Usually some cameras are left not overseen. This might delay the detection of crucial events such as property damages or even life threatening issues. To prevent such cases a video anomaly detection algorithm can be trained to detect possible anomalies to warn the overseeing personnel.

To that end, we proposed several video anomaly detection methods. First few methods use video level label information to train semi-supervised algorithms. However labeling video data is a labor intensive procedure that requires watching hundreds of hours of surveillance videos. Hence, we also proposed unsupervised anomaly detection methods that can be trained with videos without labels.

These methods use C3D video network (Tran, Bourdev, Fergus, Torresani & Paluri, 2014) to extract features. Each video is divided into 32 non overlapping segments and each segment gives 4096 dimensional features from the C3D network. For semi-supervised models these 4096 dimensional features are reduced to 4 dimensions with a semi-supervised network. Then these 4 dimensional features are used to train a future feature predicting recurrent neural networks (RNN). A neural network with multiple instance learning (MIL) and an RNN with cross entropy loss are also trained for comparison. For unsupervised methods the 4096 dimensional features are reduced to different sizes using PCA and AE, which are again used to train future feature predicting RNNs.

## 1.2 EEG P300 Dataset

Brain Computer Interfaces (BCI) provides a platform to specifically communicate via a computer system using human brain signals without any muscular activity. Due to its affordable and portable nature, EEG signals have been widely used

in many clinical and research BCI applications (Wolpaw, Birbaumer, McFarland, Pfurtscheller & Vaughan, 2002). The EEG signals can be treated as a depiction of the brain's electrical activity as measured by multiple electrodes carefully situated on the scalp. The collected waveforms are carefully combined and collectively used to obtain a real-time control signal (Mason, Bashashati, Fatourechi, Navarro & Birch, 2007) for later BCI processing by running specially designed signal processing and machine learning algorithms.

There are four different brain rhythms that can be identified in the associated frequency domain, namely delta (1-4 Hz), theta (4-7 Hz), alpha (8-12 Hz) and beta (12-30 Hz) (Klimesch, 1999). Categorically, the brain rhythms are considered as the fundamental components in the frequency domain representations of EEG signals. On the other hand, among several categories of EEG-based BCIs eliciting ERP components (brain cell responses to specific cognitive, sensory or motor events measured through EEG), the P300 potential is perhaps the most extracted and well studied component (especially for developing stable BCIs for entering texts, i.e., spellers). Upon the presentation of a visual, auditory or somatosensory stimulus in an oddball paradigm (Duncan-Johnson & Donchin, 1977), a large positive deflection (so called P300 potential) is produced in the EEG signal and measured typically around the parietal lobe nearly at 250-500 ms after the onset (Squires, Squires & Hillyard, 1975). The overall mission of the P300-based BCI systems is to develop genuine detection algorithms to capture and characterize such changes in the EEG signal and finally establish an appropriate control channel.

The P300 potential has an amplitude typically in the range of 2 to 5 $\mu$V with a duration of 150 to 200 ms (Polich, 2007) as this can be quite low compared to background brain activity and would require exclusive signal processing and machine learning techniques. Also, it might not be clear how to deal with and combine the multiple-channel signal outputs in a coherent way. One of the common approaches (examples can be found in (Kachenoura, Albera, Senhadji & Comon, 2008) and (Pires, Castelo-Branco & Nunes, 2008)) is to use the ensemble average of EEG signals over multiple-channel responses to the same stimuli in order to enhance P300 response for better identification of the specific stimulus (e.g. target character in the case of a speller (Spüler, 2017)) through P300 detection while suppressing background EEG activities as much as possible.

BCI spellers are one of the most well-known applications of the P300 response. The row/column (RC) paradigm is used in these spellers to detect the target letters to be spelt one by one by the user. In this paradigm, a matrix (e.g. $6 \times 6$ or $5 \times 8$) of characters is visually presented on the screen while rows and columns

are randomly flashing for stimulation (see Fig. 3.3). Upon the user focusing on the target character that they intend to spell, flashing rows and columns evoke the P300 potential response (embedded in the EEG signal along with other background activity and noise) each time the target is hit. This classical RC paradigm for generating flashing patterns in P300-based BCI speller was originally introduced by Farwell and Donchin in 1988 (Farwell & Donchin, 1988). Although there are other paradigms such as single character (SC) ( C Guan , Thulasidas & J Wu , 2004) or lateral single character(Pires, Nunes & Castelo-Branco, 2012a), and different region based (Fazel-Rezai & Abhari, 2009; Oralhan, 2019) or checkerboard paradigms (Townsend, LaPallo, Boulay, Krusienski, Frye, Hauser, Schwartz, Vaughan, Wolpaw & Sellers, 2010) in the literature, the RC paradigm is shown to be generally more robust to noisy measurements (Pires, Nunes & Castelo-Branco, 2012b). Several other paradigms even combine visual stimuli with audio stimuli to help augment the detection process and eventually increase the detection rate (Belitski, Farquhar & Desain, 2011) which, however, would not apply to people with major hearing impairments.

The application area of P300 response is not limited to spellers; it is indeed abundant. Similar to other BCI paradigms, P300-based paradigms can be used for controlling a wheelchair (Eidel & Kübler, 2020; Rebsamen, Burdet, Guan, Zhang, Teo, Zeng, Ang & Laugier, 2006) and helping disabled individuals for rehabilitation (Daly & Wolpaw, 2008; Duvinage, Castermans, Petieau, Seetharaman, Hoellinger, Cheron & Dutoit, 2012). It can also assist interacting with other individuals through gaming interfaces (Kaplan, Shishkin, Ganin, Basyul & Zhigalov, 2013; Rohani, Sorensen & Puthusserypady, 2014).

A goal of the presented thesis is to generate a new P300-based BCI speller EEG dataset in the RC paradigm (since the RC paradigm is more robust to noisy measurements (Pires et al., 2012b)), and introduce it to the public use of BCI and anomaly detection research alike. We emphasize that the main technical objective of P300-based BCI spellers is to decode the target characters fast and accurately through classification algorithms. To train these algorithms, a large dataset with fast stimulus presentation is certainly required. However, the current datasets in use have either an insufficient number of subjects (yielding limited amounts of data that in turn detrimentally affects the training) or a long stimulus duration (SD) (leading to impractically slow spelling). In order to generate a far larger scale and faster speller data, we conducted EEG experiments with significantly more number of participants, where we used shorter flashing and inter stimulus interval (ISI) compared to the commonly used and known datasets (Blankertz, Muller, Curio, Vaughan, Schalk, Wolpaw, Schlogl, Neuper, Pfurtscheller, Hinterberger, Schroder & Bir-

baumer, 2004; Blankertz, Muller, Krusienski, Schalk, Wolpaw, Schlogl, Pfurtscheller, Millan, Schroder & Birbaumer, 2006). In particular, inspired by the SSVEP-based speller studies (cf. Wang, Chen, Gao & Gao (2017)), we used a $5 \times 8$ character matrix to display 40 characters all at once. During the stimulation period for each character recognition epoch, every row and column is flashed 15 times. In this setting, we generated 32-channel EEG data of 18 subjects who are not specially trained for the experiment. More importantly, for fast spelling, we have used 66.6 ms SD (or intensification or simply flashing) and 33.3 ms ISI. Therefore, with the introduced dataset, one can train robust classifiers (P300 detectors) while also addressing the challenges of fast stimulation, which could potentially help to develop more practical real-life spellers. Moreover, a statistically more reliable benchmarking can be obtained for comparing various decoders. Our dataset can be accessed for public use at the link given in Abstract.

## 1.3 Thesis Organization

In this thesis five chapters are presented including this Chapter 1. In Chapter 2 related past studies are presented. In Chapter 3 detailed explanations of proposed anomaly detection methods and details and specifications of the proposed dataset are given. In Chapter 4 results from proposed methods and experiments with the proposed dataset are given. In Chapter 5 conclusions and further research directions are mentioned.

# 2. Related Work

The purpose of the anomaly detection is to find events that do not adhere to the usual patterns. Due to this property anomaly detection can be employed in many different fields to improve performance or reduce cost. It can be used to detect frauds, for instance in money transactions (Pourhabibi, Ong, Kam & Boo, 2020), check quality of the products and find defective products in manufacturing (Stojanovic, Dinic, Stojanovic & Stojadinovic, 2016). Anomaly detection can also be used in cybersecurity (Ten et al., 2011) to detect network attacks.

Images and videos are common subjects of anomaly detection. Tampering detection in images (da Costa, Papa, Passos, Colombo, Ser, Muhammad & de Albuquerque, 2020), accident detection in traffic videos (Doshi & Yilmaz, 2020) and abnormal human behaviour detection in recordings of crowded places (Yuan, Fang & Wang, 2015), are a few examples.

Besides engineering, anomaly detection has many applications in medical practices as well. It can be used to monitor individuals with health problems (Sodemann, Ross & Borghetti, 2012), it can be used to detect anomalies at medical imaging which indicates certain health conditions (Tschuchnig & Gadermayr, 2021). Coronavirus from chest X-rays (Kim et al., 2021) and dementia from brain fMRIs (Kuo & Davidson, 2016) can be detected using anomaly detection methods. Anomaly detection is also applied to medical time series such as EEG (Tsiouris, Pezoulas, Zervakis, Konitsiotis, Koutsouris & Fotiadis, 2018) and ECG (Li & Boulanger, 2020).

We focused on semi-supervised and unsupervised anomaly detection in surveillance videos and publishing a new large public medical anomaly detection dataset. In Section 2.1 a brief literature review for anomaly detection in surveillance videos and in Section 2.2 a brief literature review for P300 based BCI spellers are presented.

## 2.1 Anomaly Detection in Surveillance Videos

Anomalies are defined as abnormal and criminal behaviors in the context of the surveillance videos. Some of the examples of the anomalies in the surveillance videos are road accidents, robberies and fights. Goal is to separate videos containing anomalies from normal videos. Several studies also specialize in finding the anomalous frames or classify the type of the anomaly.

There are two types of surveillance videos used in anomaly detection studies. Many studies use videos from the same camera or a few cameras showing the same scene from different angles (Mahadevan, Li, Bhalodia & Vasconcelos, 2010) while other studies use a lot of different videos from different cameras (Sultani et al., 2018). The second type of studies include videos with different backgrounds, combining traffic surveillance, streets, crowded squares and indoor halls. In this thesis UCF-Crime dataset is used which falls into the second category and will be explained in detail in the Chapter 3 (Luo, Liu & Gao, 2017; Sultani et al., 2018). The datasets mentioned in this chapter are listed and their unique properties are highlighted below.

- ShanghaiTech dataset contains 437 videos taken from 13 different scenes and contains pixel level labels for anomalies (Luo et al., 2017).
- Another dataset contains videos from 8 different cameras overlooking crowded mostly indoor spaces. This dataset is occasionally referred to as the Subway dataset due to most studies using videos from cameras at a subway exit and entrance (Adam, Rivlin, Shimshoni & Reinitz, 2008).
- UCSD dataset is taken by a static camera recording civilians walking in a walkway. Anomalies are either non-pedestrians in the walkways or unexpected crowd behaviours. Dataset is divided into two subsets called Ped1 and Ped2 having 34 and 16 videos respectively (Mahadevan et al., 2010).
- The BOSS dataset contains 14 videos from 9 different cameras on a train carriage. Some anomalies include assault, falling, stealing (Wang & Xia, 2019).
- Avenue dataset includes 15 videos each takes approximately 2 minutes. Videos are taken by a stationary camera with 150 frame per second (Lu, Shi & Jia, 2013).
- Street Scene is a relatively new dataset containing 81 videos from a single street taken from the same position. It only contains naturally occurring anomalies. All videos are taken at daytime in the summer (Ramachandra & Jones, 2019).

Most anomaly detection studies using surveillance videos are based on statistical methods, reconstruction based methods or prediction based methods. (Santhosh, Dogra & Roy, 2020). Reconstruction and prediction based methods use neural

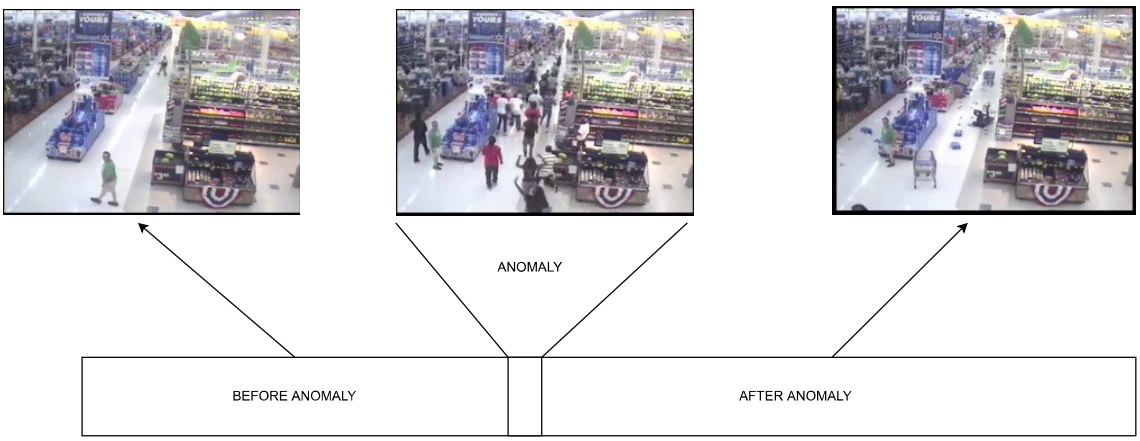



**Figure 2.1** Typical structure of an anomaly video in UCF-Crime dataset. Anomalous sections only constitute a small part of the video

networks. Training neural networks requires large amounts of data, and labeling surveillance video data requires human workforce, where annotating anomaly start and end points requires even more work. A summary of the prominent literature datasets is as follows.

Supervised learning is the general name of methods that learn classification by using clearly labeled data; in this technique every training instance has a label (Schapire & Freund, 2012). In surveillance videos, a video can be normal or abnormal; however as in Figure 2.1, most parts of an abnormal video are normal with only a few sections containing anomalies. Supervised learning is done by labeling each frame of video as normal or abnormal.

One of the most common supervised anomaly detection methods is to only use normal videos as training data (Sodemann et al., 2012). This approach removes the need of annotating anomalous frames in the abnormal videos. Also abnormal videos are much rarer than the normal videos so this method makes it easy to gather a large training set. In one of these studies motion vectors from each frame is found. Then, each frame is segmented and motion vectors of a segment are concatenated to obtain feature vectors. These feature vector dimensions are reduced using principal component analysis (PCA) or locality preserving projection (LPP) and extracted features are used to form a subspace. Lastly, a gaussian mixture model (GMM) is trained to detect anomalies. Subway dataset is used in this study (Tziakos, Cavallaro & Xu, 2010).

In another research using only normal videos to train, traffic videos are segmented into spatiotemporal video volumes and these segments are used to train a denoising autoencoder (AE). Autoencoder features are used to train a one class support vector machine (SVM). Anomaly scores are obtained by combining output of one class SVM

and reconstruction loss of autoencoder. To decide if an anomalous event happened they also tracked trajectories of the vehicles and checked if there was an intersection. For this study, a traffic surveillance dataset is collected and made publicly available (Singh & Mohan, 2019).

Another study uses an adversarial network to learn normal videos. They extracted features from each frame using an encoder and used a bidirectional convolutional long short-term memory (LSTM) to predict current frames features from past and future frames and reconstructed the current frame and used a discriminator to separate predicted frames from real one. All neural networks are trained together. UCSD and Avenue datasets are used in this study (Lee, Kim & Ro, 2018).

One study combined supervised learning with generative models, a generative adversarial network (GAN) structure called pix2pix-CGAN, is used to encode current frames and predict future frames. This model included an encoder, a decoder and a discriminator. Prediction error is used as a metric for unsupervised learning but also prediction error maps are classified by an SVM to classify anomalous frames. With this method, also in which parts of the frame anomaly are happening can be found. UCSD, Avenue, Subway and BOSS datasets are used in this study (Vu, Boonaert, Ambellouis & Taleb-Ahmed, 2021).

Anomaly videos can be classified using MIL. In this method instead of labeled instances, we have labeled bags containing many unlabeled instances (Dietterich, Lathrop & Lozano-Pérez, 1997). In video anomaly detection a video would be a bag and each frame or segment would be an instance. Anomaly videos are positive labeled bags which contain both positive and negative instances while normal videos are negative labeled bags containing only negative instances (Sultani et al., 2018). This approach eliminates the need for labeling each instance thus enabling many semi-supervised methods.

Another study keeps the segmentation method same and changes the feature extraction method to a pretrained 3D-RESNET 34 (Hara, Kataoka & Satoh, 2017) while basing the MIL loss on highest 3 anomaly scores in a positive bag, instead of the maximum score (Dubey, Boragule & Jeon, 2020; Sultani et al., 2018).

A study considered video level labels as noisy labels and employed a noise cleaning network to find at what parts of the videos the anomalous events are happening. They used C3D and TSN networks to feature extract.Then a feature similarity graph and the idea that anomalous frames should proceed each other are used to correct the faulty labels. UCF, ShanghaiTech, UCSD datasets are used in this study (Zhong, Li, Kong, Liu, Li & Li, 2019).

## 2.2 EEG P300 Dataset

Although many P300 detection and classification studies have been published in the context of P300-based BCI-based spellers, most of them utilized the publicly available BCI competition II (IIb) (Blankertz et al., 2004) and BCI competition III datasets (Blankertz et al., 2006). In both of these experimental studies and data collection processes, the speller interface was based on a $6 \times 6$ character matrix using only 3 subjects. In addition, an intensification (SD) lasts 100 ms and there is 75 ms for the ISI between two successive flashes. For each character, the corresponding row and column of the character matrix are intensified/flashed only once and randomly, and this process repeats itself typically 14 more times resulting in 180 intensifications in total per character in the matrix. Despite that these two datasets have their own challenges and peculiarities, many studies tested these waveforms quite thoroughly in the literature and reported near perfect results in the past two decades. These datasets were traditionally generated for competition purposes. However, after the competition took place, many supervised and unsupervised machine learning techniques are implemented and tested on these datasets. Accordingly, almost 100% decoding accuracy in Dataset II and 99% accuracy in dataset III are reported (Cecotti & Graser, 2011). Also in few subsequent studies, remarkable decoding accuracies are shown to be possible using only the first five or ten repetitions of the experiments (Kindermans, Verstraeten & Schrauwen, 2012; Mirghasemi, Fazel-Rezai & Shamsollahi, 2006). We consider that these near perfect (i.e. saturated) decoding accuracies indicate that these datasets are already well explored and perhaps not sufficiently representing all the challenges of a speller in daily life practices. Moreover, an accuracy figure shown by using only 3 subjects might be unreliable considering the strong EEG variability from one person to another. For that reason, in our dataset, we have used faster stimulation (SD: 66.6 ms and ISI: 33.3 ms in our case vs SD: 100 ms and ISI: 75 ms in theirs) that significantly hardens the decoding task because faster flashes causes strong P300 interference, which -on the other hand- represents the conditions of a desirable fast speller more accurately. This would immediately degrade the decoding accuracy of the state-of-the-art classifiers and in turn require more sophisticated approaches to push it back up. Also, we have experimented with 18 subjects which is a far larger set than that (3 subjects only) of these competition datasets.

We have also identified another publicly available and relatively recent dataset in the literature that considers a larger number of subjects (Lee, Kwon, Kim, Kim, Lee, Williamson, Fazli & Lee, 2019). Although 54 subjects have participated in their

Table 2.1 Comparisons between the existing P300 speller datasets and ours.

| | # of subjects | # of chracters spelled | # of classes | ISI | SD | # of repetitions |
|---|---|---|---|---|---|---|
| Our dataset | 18 | 160 | 40 | 33.3 ms | 66.ms | 15 |
| BCI COMP II Blankertz et al. (2004) | 1 | 73 | 36 | 75 ms | 100 ms | 15 |
| BCI COMP IIIBlankertz et al. (2006) | 2 | 180 | 36 | 75 ms | 100 ms | 15 |
| OpenBMILee, Won, Kwon, Jun & Ahn (2020) | 54 | 69 | 36 | 135 ms | 80 ms | 5 |
| Hoffmann Hoffmann, Vesin, Ebrahimi & Diserens (2008) | 9 | 135 | 4 | 300 ms | 100 ms | 1 |

experiment and the speller interface was again based on a $6 \times 6$ character matrix, their paradigm is not the RC paradigm and their experiments are shorter than ours. In their study, each character is flashed individually and SD is set to 80 ms while ISI between flashes is set to 135 ms, resulting in a too slow stimulation which surely does not address the need for fast spelling in practice. Note that the stimulation in our introduced dataset is at least two times faster than this dataset of (Lee et al., 2019) (SD: 66.6 ms and ISI: 33.3 ms in our case vs SD: 80 ms and ISI: 135 ms in theirs). Moreover, including multiple blocks in our study helped us to reduce the within-subject variability and assess the between-subject variability more reliably.

We emphasize that the performance of a P300-based BCI speller is typically measured by the amount of information (bits) that is transferred per unit time from brain to computer through the interface. This information transfer rate increases as the decoding accuracy increases, but it decreases as the SD increases. Hence, the goal of the researcher is to increase the accuracy and decrease the SD. The near perfect accuracies in the literature clearly show that the attention must be given to decreasing the SD, and to our best knowledge, no past dataset studies seem to take into account this. We conducted the presented study to address this need by using stimulation that is at least twice faster compared to the literature. On the other hand, faster stimulation leads to more severe P300 signal interference, and in turn degrades the decoding accuracy which can be met by more sophisticated signal processing and machine learning algorithms especially given the recent advancements in the field of deep learning. Hence, we believe that the right direction in this field to speed up the stimulation and fight back the challenges due to the resulting P300 interference via sophistication in the processing. Our study and the introduced spelling dataset (with fast stimulation) can be seen as a step towards this goal. In addition, we present data from 18 subjects which is larger than most studies, and we also used 40 ($5 \times 8$) characters instead of the typical choice of 36 ($6 \times 6$) which is another step at increasing the information transfer rate and more in line with the aspect ratio of most monitors. Unlike publicly available aforemen-

tioned datasets (Blankertz et al., 2004,0; Lee et al., 2019), plenty of past studies present P300-based BCI speller experiments with inaccessible datasets in which different paradigms besides the RC paradigm have been compared. For example, SC (intensification of a single character) and RC paradigms are compared ( C Guan et al., 2004). Similarly, in (Fazel-Rezai & Abhari, 2009), comparisons of the RC paradigm and region-based P300 paradigms are presented. Certain other studies focus on the structure of the character matrix such as a dictionary-driven character order instead of an alphabetical order (Ahi, Kambara & Koike, 2011), or focus on the effects of background character size and color on the character decoding accuracy (Salvaris & Sepulveda, 2009). Further experiments are conducted for an audience of P300-based BCI spellers with ALS patient-participants (Guy, Soriani, Bruno, Papadopoulo, Desnuelle & Clerc, 2018; Nijboer, Sellers, Mellinger, Jordan, Matuz, Furdea, Halder, Mochty, Krusienski, Vaughan, Wolpaw, Birbaumer & Kübler, 2008; Sellers, Kubler & Donchin, 2006). The influence of the eye gaze movements on the performance is also considered to investigate the various clinical applications of P300-based BCI spellers (Brunner, Joshi, Briskin, Wolpaw, Bischof & Schalk, 2010). Few more studies have conducted experiments using alphabets of other languages besides English (Chaurasiya, Londhe & Ghosh, 2016; Kabbara, Hassan, Khalil, Eid & El Falou, 2015; Minett, Zheng, Fong, Zhou, Peng & Wang, 2012). The datasets of these studies are not publicly available and in none of those the focus is on the algorithms of decoding, which is in sharp contrast to our goal in this study.

Two of the past studies focus on finding the best EEG channels for P300 detection by using two datasets in which 18 and 6 subjects are involved, c.f. (Colwell, Ryan, Throckmorton, Sellers & Collins, 2014) and (Krusienski, Sellers, McFarland, Vaughan & Wolpaw, 2008), respectively. In addition, studies such as (Krusienski, Sellers, Cabestaing, Bayoudh, McFarland, Vaughan & Wolpaw, 2006) compare classification techniques for P300 detection using that dataset of 6 subjects, while (Krusienski, Sellers & Vaughan, 2007) investigates to find common patterns for P300 signals using speller settings with 7 subjects in their experiment. Although most of these experiments are conducted under controlled environments, subsequent experiments that take place in noisy environments achieved very close levels of accuracy to that of noiseless case (Ortner, Prueckl, Putz, Scharinger, Bruckner, Schnürer & Guger, 2011). This demonstrates that these speller designs are quite suitable for everyday applications. An intuitive conclusion is made in (Lu, Speier, Hu & Pouratian, 2012) that it is possible to increase ITR by decreasing both SD and ISI. Additionally, an interesting outcome of this study is that models trained with a specific SD and ISI can be used to classify data obtained with different ISI and SD selections, i.e., even when unmatched with the training settings (Xue, Tang, He, Xu

& Qi, 2019). Therefore, considering the recent developments on signal processing and machine learning techniques on a par with the available computational power, published standards can be achieved with a faster stimulus timing which eventually results in more challenging datasets.

All the studies mentioned in this section either use the two competition datasets (Blankertz et al., 2004,0) or their own dataset which are not made publicly available (except (Lee et al., 2019) as explained above), for investigating specific aspects of P300-based BCI spellers.

Contrary to those datasets, the main purpose of this study is to supplement a large and more challenging P300 dataset (with shorter SD and ISI) to help to improve the speller information transfer speeds.

## 2.3 Novel Contributions and Highlights of the Thesis

- We proposed a versatile video anomaly detection method where feature extraction and anomaly detection can be trained separately. These methods both have semi-supervised and unsupervised variants.
- The proposed methods use RNNs to exploit temporal relations between concurrent video frames.
- Only a small portion of the anomalous video contains anomaly, the semi-supervised methods use MIL loss to incorporate this knowledge. The unsupervised methods try to predict future frames from current frames. These methods cannot predict anomalous parts of the video because anomalies are unexpected. Our method works with a limited number of frames (ranging from 4 to 16). Even though we cannot reconstruct frames from predicted features, the limited number of frames provide us easy and quick to train models.
- In order to facilitate development of new time series anomaly detection methods, we also introduced a new, large, publicly available P300 speller dataset. This dataset contains EEG recordings of 18 people spelling 160 characters. Considering the number of subjects and amount of characters spelled, this dataset is the largest and most challenging P300 speller in the literature.
- The introduced dataset has shorter ISI and SD values compared to other pub-

licly accessible datasets, this makes our dataset challenging and encourages the development of better methods.

- We provided data visualization and benchmark results for the proposed dataset. We chose 3 methods from literature that are well performing 0n the previous datasets and applied them to our dataset for future comparisons.

# 3. METHODS

We proposed two new anomaly detection methods. Also an extensive anomaly detection dataset is made public. In this chapter details of the proposed methods and details of the introduced dataset are given. Section 3.1 introduces the models used in anomaly detection, feature extraction techniques. Section 3.2 gives details about the proposed dataset, its specifications, obtaining process and equipment.

## 3.1 Anomaly Detection Algorithms

We used the UCF-Crime dataset (Sultani et al., 2018) to develop our anomaly detection algorithms. This dataset consists of 1900 videos ranging from 30 seconds to 8 hours. Total run time of the dataset is 128 hours. Videos are taken from CCTV recordings on YouTube and LiveLeak. All the events are real. Videos have different frame sizes. There are 13 different types of anomalies represented in the dataset: abuse, arrest, arson, assault, road accident, burglary, explosion, fighting, robbery, shooting, stealing, shoplifting, and vandalism. Training set consists of 1610 videos, 800 normal and 810 anomalous videos. Test set consists of the remaining 290 videos, 150 normal and 140 anomalous videos.

One of biggest challenges of the working with videos, especially UCF-crime dataset, are the size of the data, to overcome this challenge most studies use either extracting features from each frame (Zhong et al., 2019) or dividing videos into segments and extracting features from each section, via pre-trained networks. We used C3D (Tran et al., 2014) as suggested by the dataset publishers. C3D takes 16 consecutive frames as an input and classifies videos by using 3 dimensional neural network layers. C3D used in this thesis is trained on the Sports1-M (Karpathy, Toderici, Shetty, Leung, Sukthankar & Fei-Fei, 2014). UCF-crime videos are divided into 32 equally sized segments with no overlap. Each 16 consecutive frame segment is plugged into the

C3D network and a feature vector of size 4096 is obtained from the fc6 layer of the network. Each video is resized to $240 \times 320$ size so it can be compatible with C3D input size. Mean of each feature vector belonging to a segment is taken as the feature vector of that segment, then L2 normalization is applied to feature vectors, so regardless of video duration or frame rate each video is summarized by 32 features vectors (each being 4096 dimensional). A simple illustration of the C3D model and feature extraction technique can be found in Figure 3.1. As previously mentioned video duration in the dataset differs greatly, all videos are processed to have 32 segments to prevent longer videos to be dominant in the network training, however this process reduces information contained in the longer videos.

In the following subsections our semi-supervised anomaly detection method using MIL loss and unsupervised technique to predict features of the future frames, are explained.

### 3.1.1 Semi-Supervised Video Anomaly Detection

In this section we propose 3 novel anomaly detection methods to detect anomalous videos and frames using features obtained from C3D. Our motivation is to exploit the fact that only a small part of the anomalous video contains anomalies. Also we wanted to train an RNN to learn temporal connections between concurrent C3D features. To this end we trained a neural network with an MIL loss to reduce feature dimensions of C3D features. And using these features we trained two RNNs.

First we train a deep neural network with encoder properties. First 2 layers drop 4096 dimensional features to 512 and 128 dimensions respectively. These layers are trained with both reconstruction loss and classification loss. After the second layer, every segment is concatenated with the next 2 segments which are called windows in the context of this thesis. For example the 1st window is the 1st, the 2nd and the 3rd segments concatenated, the 2nd window is the 2nd, the 3rd and the 4th segments concatenated and the 30th window is the 30th the 31st and the 32nd segments concatenated. The 2nd layer network continues with 384 dimensional features and 30 windows. After forming the windows network reduces 384 dimensional features to 64, 32 and 4 in 3 layers. All layers except the last one have 0.6 dropout rate and ReLu activation. The network is illustrated in the 3.1.

The last layer is a classification layer with the loss function in equation (3.1). In this equation $V_i$ is the $i$th video. $y_i$ is label belonging to $i$th video (0 if video has no

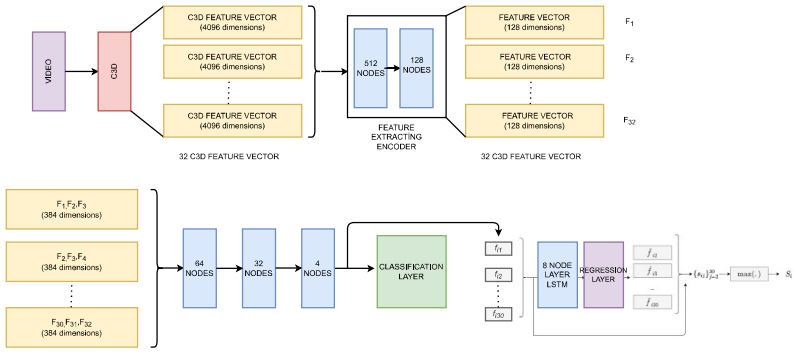



**Figure 3.1** Semisupervised model

anomaly, 1 if the video has anomaly), and $a_{ij}$ is the score belonging to $j$th window of the video. The whole network is back-propagated with this loss function. The purpose is to use the knowledge that normal videos have no anomalies and videos with anomalies have only a small section including the said anomaly. Loss function is only affected by the window with highest anomaly score, if the video is normal even the highest scoring window should have a low anomaly score, while videos with anomalies should have at least one window with anomaly and the window with the maximum score is assumed to include an anomaly. So loss is a modified version of the cross entropy loss

$$l(V_i) = -y_i \log(\max_j(a_{ij})) - (1 - y_i)(\log(1 - \max_j(a_{ij}))) \tag{3.1}$$

Training is done with the training part of the dataset consisting of 1610 videos. C3D features are fed into the mentioned network. The network is trained with the loss function defined in equation (3.1). First two layers of the network are also trained with reconstruction loss in addition to loss function (3.1). Adam optimizer (Kingma & Ba, 2014) is used with initial learning rate 0.0001. We call this structure the NN model.

For the second method NN model is used as a feature extraction method. Features are taken from the last layer before the classification layer. With this method 4096 dimensional features are reduced to 4 dimensions with a semi-supervised network. Also 32 segments are combined into 30 windows. Since each window is consecutive a recurrent neural network can be used to exploit time relationships between windows.To this end two different RNN models are proposed. The first RNN model is trained with 4 dimensional features. Training is done using cross entropy loss with video level labels. Network takes sequential inputs with size of 4. First layer is an LSTM layer with 8 hidden nodes and ReLu activation, the second layer is a

classification layer with softmax activation and cross entropy loss. Adam optimizer is used with initial learning rate 0.0001. This model is called RNN classification model.

By definition anomalous events are unexpected events. The second RNN model is proposed to exploit the idea that predicting anomalous data from normal data is harder than predicting normal data from normal data. Inputs of this RNN are the same with the RNN classification model. This model is trained to predict features of the next window from features of current and previous frames. The network is trained with the mean square error (MSE) loss. First layer is an LSTM layer with 8 hidden nodes and ReLu activation, the second layer is a regression layer with hyperbolic tangent activation function. Adam optimizer is used with initial learning rate 0.0001.This model is called RNN prediction model.The RNN prediction model and RNN classification model are illustrated in Figure 3.1.

### 3.1.2 Unsupervised Video Anomaly Detection

As mentioned in Section 2.1 anomalies are hard to annotate due to this, unsupervised learning methods are useful. With the proposed RNN classification model we obtained an unsupervised video anomaly detection method working on features obtained with a semi-supervised feature extraction method. If a feature extraction method that reduced dimensions from 4096 to 4 can be replaced by an unsupervised method then the entire process would be unsupervised. For this purpose we employed PCA and stacked AEs. Also feature sizes of 8 and 16 are considered instead of 4.

PCA is a linear transformation technique to change the basis of a space. It chooses orthonormal vectors as a new basis. First vector is in the direction of the line which makes squared distance to all points minimum and the consecutive eigenvectors are found by finding the line that has minimum squared distance to all points while being perpendicular to all previously found vectors. Since the direction of the line that has minimum squared distance to all points is the direction of maximum variance, every feature conveys more information than ones coming after them. Because of this PCA can be used to extract features without losing much information (F.R.S., 1901).

C3D features from the training set are normalized and coefficients to perform PCA are found. Using these coefficients the first 4 components are chosen as extracted

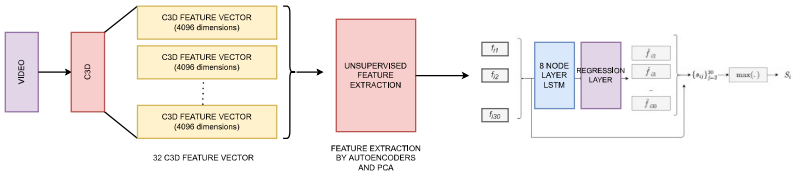



**Figure 3.2** Unsupervised model

features. These first 4 components contain the total variance. With 4 features for every video segment an RNN similar to RNN prediction model is trained.

As an alternative to PCA, a stacked auto encoder is trained. First an AE reduces 4096 dimensions to 512, then another AE reduces dimensions to 4. AE is trained with MSE loss with L2 and sparsity regularization. L2 regularization coefficient is 0.001 and sparsity regularization coefficient is 0.05. Sigmoid transfer function is used in the decoders. To use sigmoid function features are standardized with equation (4.2) where $f$ is the C3D feature matrix, $f_s$ is standardized feature matrix, $i$ is the feature number and j is the video number.

$$f_s(i,j) = (f(i) - \min_j f(i,j)) / \max_j f(i,j) \tag{3.2}$$

With the 4 features extracted from AEs, another RNN is trained. This RNN shares the same structure, parameters and training options with the RNN trained using features obtained by PCA. Both models are shown in Figure 3.2.

Its easier to train an RNN predicting future features with lower feature dimension. For this reason the feature dimension of 4 is choosen; however, if we increase the feature dimension more information can be found after feature extraction. In the semi-supervised version feature extractor was also an anomaly detection method. Because of this, 4 features were ensured to summarize anomalous parts of the video. For unsupervised method we repeated experiments with feature size of 8 and 16.

In real life, anomalies occur rarely but UCF-crime dataset has 800 normal videos and 810 anomalous videos. For our experiments to be unsupervised, we prepared two partitions of dataset one with 1% of the anomalies and another with 10% of the anomalies. The entire dataset and only normal videos are also used as benchmarks. Anomaly videos used in partitions are chosen randomly.

## 3.2 Proposed Dataset

We present our dataset in this section, and also explain the conducted EEG experiments together with the associated equipment. Below, we begin with the stimulus presentation.

### 3.2.1 Stimulus Presentation

In our P300-based BCI speller EEG experiments, we use a $5 \times 8$ (26 English alphabet characters, 10 digits, 3 punctuation marks and the backspace) character matrix to display 40 characters all at once, each of which occupies an equal size on the screen. An ASUS XG258Q LCD (Liquid Crystal Display) monitor with $1920 \times 1080$ resolution and 60 Hz refresh rate is utilized to present the character matrix to the subjects. The viewing distance between the participant and the screen is fixed to an average of 57 cm in all of the conducted experiments. The height and width of the characters is set to 80 pixels and 70 pixels, respectively which makes visual angle of a single character $2.2^\circ \times 1.9^\circ$. On top of the character matrix, a dark gray rectangular row is placed and the target words are configured to appear on this rectangle during the training phase. A simple visual illustration of the experiment can be found in Fig. 3.3. The stimulus presentation software is written in MATLAB (Mathworks, Inc.) utilizing the Psychophysics Toolbox (Version Number 3.0). All the characters are originally painted in gray color, and when a row or column flashes, it highlights all the characters that belong to the corresponding row/column. The highlighting is realized by making the luminance of characters turn from gray to white for the duration of the stimulation time and the character fonts turn into bold type such that their occupied area on the display grows in order to maintain the subject's attention.

### 3.2.2 Experimental Set-up

Our experiments are cue-guided. Namely, the target word first appears in the middle of the screen before the character matrix is presented on the display, and it stays still for a duration of two seconds. For each subject in the experiment, 10 blocks of word bundles (cf. Table 3.1) are presented where blocks include 2 to 3 words per each. Words in the bundle are carefully selected such that there are enough and roughly balanced numbers of flashes for each row and column. The exact number

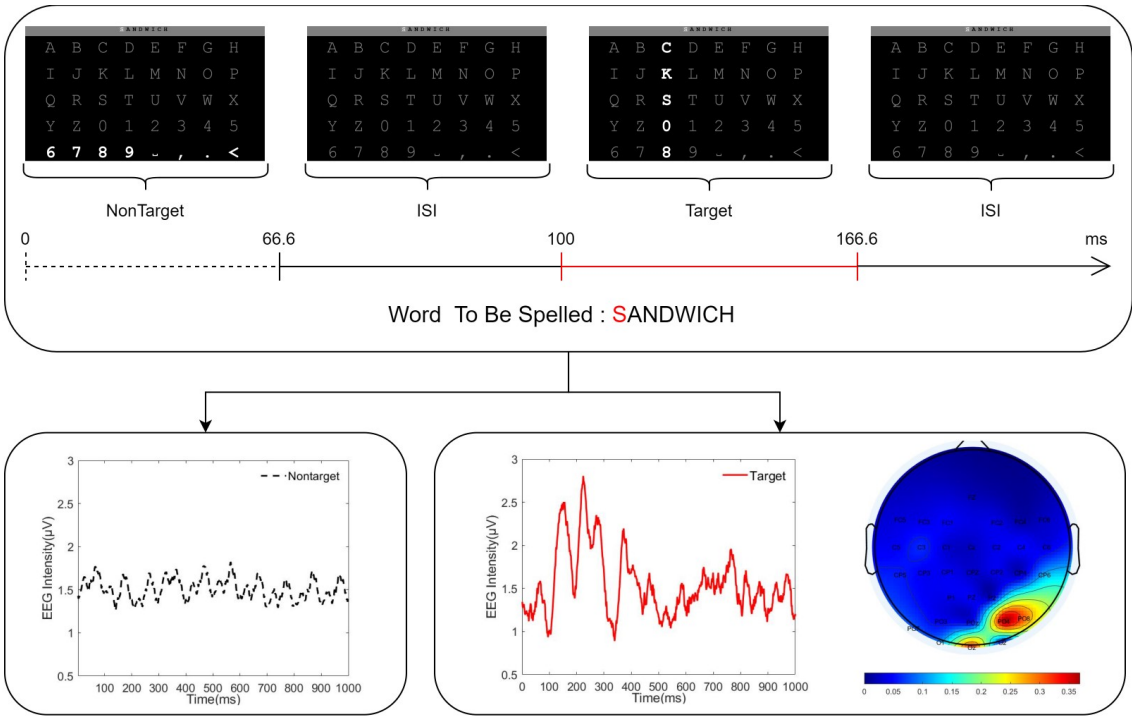



**Figure 3.3** The speller interface used in our experimental set-up is shown. It consists of 26 letters, 10 digits, two punctuation marks (?,.), a space (⎵) and a backspace character (<). On the top of the figure, a short example of the timeline is shown where the target character is 'S'. On the bottom left part of the figure, the average of the non-target signals belonging to the eleventh subject is presented. On the bottom right of the figure, the average of the target signals (P300) and the corresponding topography belonging to the same subject are presented.

**Table 3.1** The duration and content of each block in our experiments.

| Block | WORD 1 | WORD 2 | WORD 3 | DURATION(sec) |
|---|---|---|---|---|
| 1 | SANDWICH | WATER | MONEY | 394.5 |
| 2 | EXACT | TIME | 17.36 | 306.5 |
| 3 | FLOWERS | ROSE | TULIP | 350.5 |
| 4 | DOGS | ARE | FRIENDS | 306.5 |
| 5 | DATE | 27.06.1935 | 7 PM | 390.5 |
| 6 | WHITE | YELLOW | PINK | 328.5 |
| 7 | MATH 101 | PSY 350 | | 328 |
| 8 | PASSWORD? | X8JVZ4Q | | 350 |
| 9 | BRAIN | P300 SPELLER | | 372 |
| 10 | HOW MUCH? | 6.75 EUR | | 372 |

appearances of each character as the target as well as the number of column and row flashes in our experiments are presented in Fig. 3.4. Depending on the number of characters in each word, words are divided into a special arrangement of blocks such that the lengths (actual duration in the experiment) of these blocks are similar.

During the experiment, a word is primarily presented in the middle of the screen before it is moved to the top. The target character in the word (after it is moved to

the top) is surrounded by a white square while the remaining characters are painted in a dark (black) color. Hence, this describes the cue we use. There is a 2.5-second delay before the actual stimulation begins. During the stimulation period for each character recognition epoch, every row and column is randomly flashed for 15 times (i.e. this process is repeated 15 times for each character, known as repetitions.). Every flashing lasts for $66\frac{2}{3}$ milliseconds and there are $33\frac{1}{3}$ milliseconds between two consecutive flashes. After each character recognition epoch, there is a 2.5-second pause time before continuing with the presentation of the next character. For a quarter of this time, the screen goes blank (a black screen) and in the remaining time, the character matrix and the word on top appears on the screen again. At the top of the screen, the new target character in its place in the word turns into white color (illuminated) and is surrounded by a white square while the other characters in the word turn black. This change (cue) is to be followed by the subject (as instructed before the experiment starts) during the short pause time between the characters. After receiving the cue in the pause time, the participant focuses on the target character in the matrix during the complete period of all 15 flashes (as also instructed before the experiment starts). At the end of the word recognition phase, the screen goes blank again for a duration of three seconds and then the next word appears in the middle of the screen again. If the last word inside the block is finished, then an instruction appears on the screen to guide the participant for a break. Whole experiment takes approximately 65-70 minutes including the break times and 58.3 minutes without counting the breaks.

### 3.2.3 Data Acquisition

In our experiments, we used a P300-based RC paradigm speller to record EEG signals from the participants, each spelling 160 characters. In total we collected EEG signals from 21 participants (8 males, 13 females). The data of the three participants are excluded from the dataset due to low decoding accuracy (measured to be in the range of 10% using any number of repetitions and methods). The participants either have normal or corrected-to-normal (near-normal) vision. An appropriately sized electrode cap (small, medium, or large) is placed on the subject's head. They sit 57 cm in front of the display monitor in a dark room (only light source is the display monitor). All of the participants are asked to provide their written consent after they are informed about the goal of the study and experimentation process. Our experimentation set-up and data acquisition processes are approved by the Research Ethics Committee of Sabanci University, Turkey. The EEG system

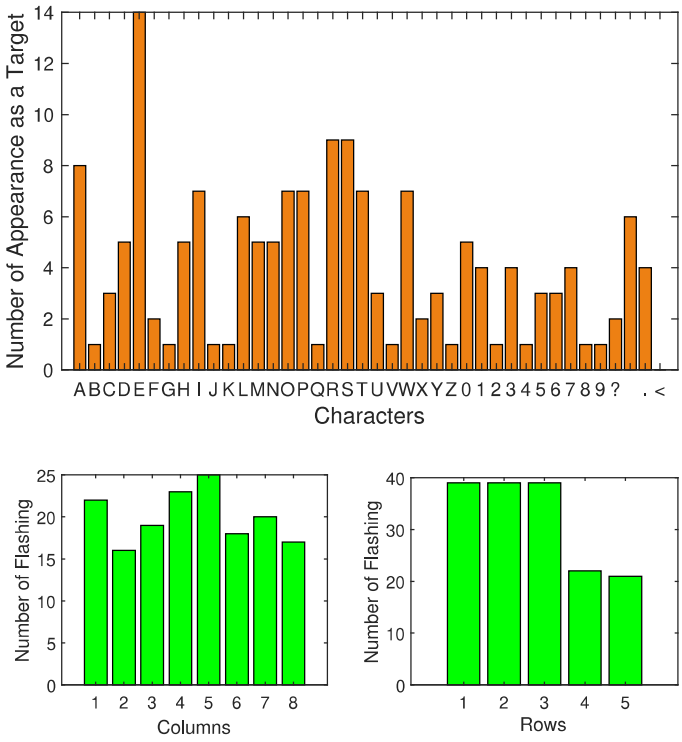



**Figure 3.4** Counts of the target characters, rows and columns in our dataset are shown.

Brain Products (Acti 64champ amplifier) (lines & planes of closest fit to systems of points in space, BP) are used to acquire the EEG signals. The incoming data is sampled at a frequency of 1000 Hz mode and no notch filter is applied during the data acquisition phase. ActiCAP slim 32-channel Ag/AgCl electrode set (by Brain Products; Munich, Germany) is used to record the EEG signals, we refer to Fig. 3.5 for the placement of the electrodes. In the same figure, the reference electrode is also shown, which is placed at the vertex (FCz) position[1]. Electrode impedances are measured around or below 10 kohms. Participants sit on a comfortable chair and are asked to remain still (to their best ability) during the data acquisition process.

### 3.2.4 70-electrode 10-10 System

We used a derivative of the international standard 10-20 system (Klem, Lüders, Jasper & Elger, 1999) which describes EEG electrode distances relative to distances

[1]In the supplementary document, we provide the electrode positions with the corresponding channels in the dataset.

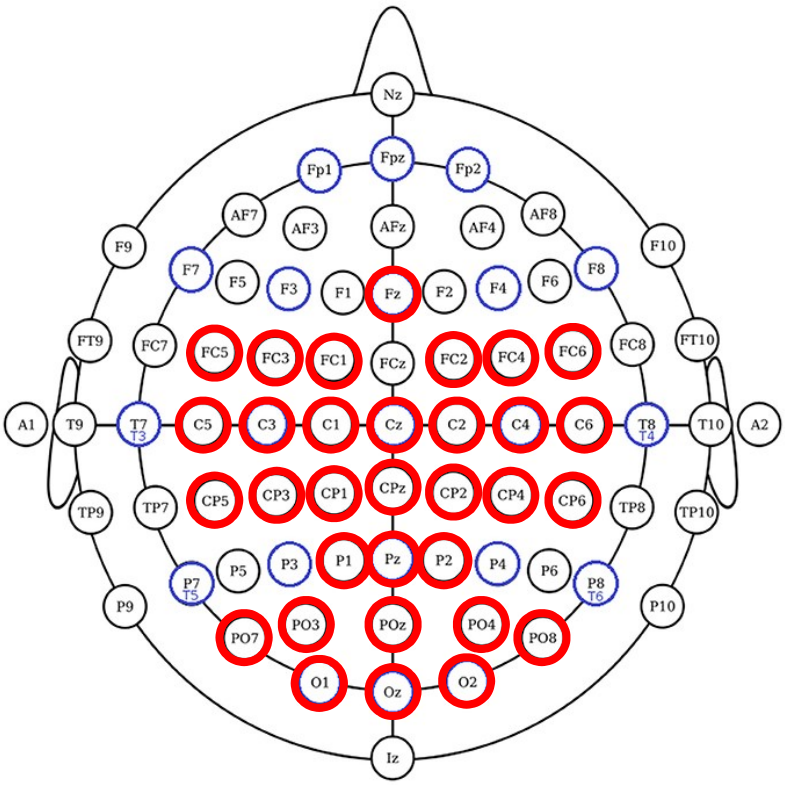



**Figure 3.5** Electrode locations of the EEG (electroencephalography) recording in our experiments are based on the 10-10 international system Nuwer et al. (1999).

between cranial landmarks. The derivative we used is known as the 10-10 system (Nuwer et al., 1999) and is obtained by making inter-electrode distances smaller. The primary purpose of all these systems is to constitute a reproducible standard for EEG electrode placement (such as 70 electrode positions in 10-10 systems as shown in Fig. 3.5). We have used locations covering parietal and occipital areas of the brain since EEG intensity is higher in these areas when evoked with a visual oddball paradigm (Ji, Porjesz, Begleiter & Chorlian, 1999). This would also allow us to make comparisons across different studies considering EEG as main means of brain functionality and also between distinct participating subjects. Channel numbers and their corresponding locations on the head cap (holder) according to the international 10-10 system is given in Table 3.2 and illustrated in Fig. 3.5. This information can alternatively be found in header metadata contained in *vhdr* files.

### 3.2.5 Access and Contents of the Dataset

We provide the dataset free of charge for the research community to use. The link below can be used to access the dataset. We would appreciate it if you could cite

| Channel Number | Channel Location | Channel Number | Channel Location |
|---|---|---|---|
| 1 | FC1 | 17 | FC4 |
| 2 | FC3 | 18 | FC6 |
| 3 | FC5 | 19 | C2 |
| 4 | C5 | 20 | C4 |
| 5 | C3 | 21 | C6 |
| 6 | C1 | 22 | CP2 |
| 7 | Cz | 23 | CP4 |
| 8 | CP5 | 24 | CP6 |
| 9 | CP3 | 25 | Pz |
| 10 | CP1 | 26 | P2 |
| 11 | CPz | 27 | POz |
| 12 | P1 | 28 | PO4 |
| 13 | PO7 | 29 | PO8 |
| 14 | PO3 | 30 | Oz |
| 15 | O1 | 31 | O2 |
| 16 | FC2 | 32 | Fz |

Table 3.2 Channel locations and corresponding channel numbers in the dataset

our work when you use our dataset in your performance analysis/research reports.

**Link**: `https://data.mendeley.com/datasets/vyczny2r4w`

Within the dataset, we reserve individual folders for each participant in our experiments. You will recognize that there are *vhdr*, *vmrk* and *eeg* files for each block in every folder where *eeg* file contains raw EEG data, *vmrk* file contains markers and their timestamps, while *vhdr* file contains header information for you to be able to open *eeg* files properly. We have run our tests in Matlab version 2019b and Letswave's (Mouraux & Iannetti, 2008) (https://www.letswave.org/) six plug-ins to open *eeg* files and get the necessary preprocessing done.

To open raw EEG data, Brainvision VHDR files should be first imported (click File → Import signals → import Brainvision VHDR files). We have used known preprocessing methods such as linear detrending (click Preproces → DC removal and Preproces → linear-detrend), segmentation (click Preprocess → segmentation/segmentation relative to events, events s1 to s13 are selected, epoch start = 0, epoch duration = 1, epoch size = 1000), FFT filtering (click Preprocess → Frequency filters → FFT filter → bandpass, low pass frequency = 0.5 Hz, highpass frequency = 30Hz) and lastly, common average reference filtering (Alhaddad, 2012) (click Preprocess → rereference, for new reference select all channels, for apply reference select all channels). We refer to each part of the preprocessed data related

to an intensification as an instance.

After the preprocessing step, EEG signal intensity averages are computed across all channels for both P300 present and non-present cases separately, and we have plotted these averages below for all participating subjects. A corresponding brain activity heat map is also provided next to these EEG signal averages. The heat maps are generated based on the weights in the feature maps of the first layer of the EoCNN method from 4 (the one using $32 \times 11$ maps). Note that for each one of the 32 channels, there is a specific feature vector of length 11, containing weights. We use the root mean square of the 11 values to obtain a final activity measure for the corresponding channel. Such channel–specific activity measures are plotted in the presented brain activity heat maps with respect to the electrode locations (with red color representing the highest and dark blue representing the lowest activities). The Matlab code for automatically generating these heat maps is given in the file "map.m" (EEGLAB 2020 is required to run this script).

As can be seen from Table 3.2, subjects showing visible separation in Fig. 4.22 (such as subjects 1, 2 and 3) also performed well with high decoding accuracy while some other subjects (such as subjects 15 and 16) did not demonstrate a similar high performance. The reason behind this can be attributed to relatively higher impedance values or perhaps unintended physical dislocations in the electrode cap during the experimentation.

# 4. RESULTS

In this chapter, results with the proposed methods and dataset are given. For video anomaly detection methods, anomaly score calculation for each method is explained, commonly used frame level and video level area under the ROC curve values are calculated and compared to other methods using UCF-crime dataset. For P300 dataset evaluation, data visualisations are provided and three commonly used methods from literature are chosen, decoding accuracy and ITR values using these 3 methods are calculated.

## 4.1 Video Anomaly Detection

The methods proposed in Chapter 3, have been applied to the test set of the UCF-crime dataset. Anomaly scores for each frame and video are found and ROC curves are drawn, also a small false positive rate is chosen to compare performance on different types of the anomalies. Areas under ROC curves are compared to similar studies and state of the art results from literature. Test set of the UCF-crime has frame level anomaly annotations so at which frames anomalies are taking place is known and can be used to determine frame level performance of the methods.

### 4.1.1 Semi-supervised Methods

For the NN model, output of the softmax layer is taken as the anomaly score belonging to corresponding windows, an anomaly score of a window is assigned to the segment in the center of the window with the exception of first and the last segments. Since these segments were not the center of any windows, first and last

window's anomaly scores are also assigned to them respectively. Since each segment consists of non overlapping frames of a video each frame in the same segment has the same anomaly score as that segment. This process is shown in Figure 3.1. The video level anomaly score is the maximum of the anomaly scores of the windows in the video. It is found by equation (4.1). $S_i$ is the anomaly score of the $i$th video and $s_{ij}$ is the anomaly score of the $i$th video $j$th window

$$S_i = max_j s_{ij} \tag{4.1}$$

With this method %81.73 video level AUC and %58.23 frame level AUC is obtained. ROC curves can be seen in Figure 4.1 and Figure 4.2. For RNN classification model,

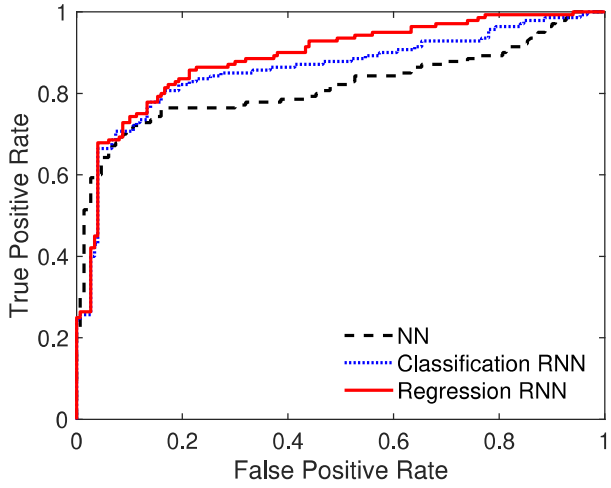



Figure 4.1 Video level ROC curve

steps of the NN model are followed. Softmax outputs are assigned to corresponding windows and window scores are backtracked to the frames and video level scores are found with equation (4.1) With this method %85.82 video level AUC and %73.71 frame level AUC is obtained. ROC curves can be seen in Figure 4.1 and Figure 4.2. For RNN prediction model from 4 dimensional features of every window, features of the next window are predicted. Squared distance between predicted features and actual features are taken as that windows anomaly score. Window level anomaly score is backtracked to the corresponding segments and frames. The video level score is the maximum of the window level scores and is found by equation (4.2).Where $s_{ij}$ is the anomaly score of the $i$th video $j$th window, $f_{ijk}$ is 4 dimensional features belonging to $i$th video $j$th window, $\hat{f}_{ijk}$ is predicted features belonging to $i$th video $j$th window and $S_i$ is the anomaly score of the $i$th video.

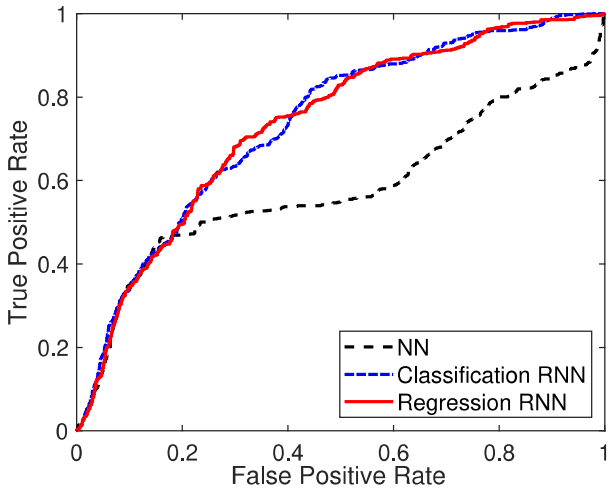



Figure 4.2 Frame level ROC curve

$$s_{ij} = \sum_{k=1}^{4} (f_{ijk} - \hat{f}_{ijk})^2 \text{ and } S_i = \max_j s_{ij} \tag{4.2}$$

With this method %88.71 video level AUC and %73.64 frame level AUC is obtained. ROC curves can be seen at 4.1 and Figure 4.2.

As 4.1 and Figure 4.2 and 4.1 shows RNN classification method performed better than the NN classification method. And the RNN prediction method performed better than both of them. This shows prediction based methods are better because they rely less on the faulty label (lack of frame level labels) and anomalies are harder to predict. However most striking result is the difference between frame level performance, even though we converted non overlapping segments into overlapping windows inside the NN to create a temporal relation NN is structure is not suitable for capturing temporal connections of the data, on the other hand RNN based models are much better at capturing the temporal relations hence finding the location of the anomaly more accurately. For anomaly classification RNN prediction model

Table 4.1 Area under the curve percentages of proposed methods and other methods using C3D are given

| METHOD | FRAME LEVEL | VIDEO LEVEL |
|---|---|---|
| NN | 58.23 | 81.73 |
| Classification RNN | 73.71 | 85.82 |
| Regression RNN | 73.64 | 88.71 |
| SULTANI et al. (Sultani et al., 2018) | 75.41 | - |
| ZHONG et al. (Zhong et al., 2019) | 81.08 | - |

is chosen since it is the best performing among proposed models. We chose 0.25

Table 4.2 Video types in the UCF dataset and results of the regression RNN on the dataset. In the first two columns true and false classification rates are given. Last column contains the number of videos for each type of the video in the test set.

| VIDEO TYPE | CORRECT CLASS | OTHER CLASSES | NUMBER OF VIDEOS |
|---|---|---|---|
| Normal | 0.78 | 0.22 | 150 |
| Arrest | 0.8 | 0.2 | 5 |
| Arson | 1 | 0 | 9 |
| Attack | 1 | 0 | 3 |
| Burglary | 0.92 | 0.08 | 13 |
| Explosion | 0.95 | 0.05 | 21 |
| Fight | 1 | 0 | 5 |
| Abuse | 0.5 | 0.5 | 2 |
| Accident | 0.87 | 0.13 | 23 |
| Robbery | 0.8 | 0.2 | 5 |
| Shooting | 0.91 | 0.09 | 23 |
| Shoplifting | 0.62 | 0.38 | 21 |
| Stealing | 0.6 | 0.4 | 5 |
| Vandalism | 1 | 0 | 5 |

false alarm rate and found corresponding anomaly score threshold. According to this threshold videos are classified and correct classification rates are given at Table 4.2. According to these results, the proposed method can detect sudden and big anomalies like explosions, road accidents and arson better than subtle anomalies like abuse or anomalies like shoplifting that can easily be confused with everyday actions.

### 4.1.2 Unsupervised Anomaly Detection

For feature extraction with PCA transformation weights are found using the training part of the dataset. Then, these weights are applied to the test set. First 4, 8 and 16 features are chosen and processed with the RNN trained on PCA features from training data. Since this model is the similar to RNN prediction model(only difference is the unsupervised feature extraction method), anomaly scores are found with the method used in equation (4.2). This time $f_{ijk}$ is features from the PCA.

Features obtained from AE are used in the same way with the PCA features. These experiments repeated with 8 dimensional and 16 dimensional features and different

Table 4.3 Area under the ROC curve values for each proposed unsupervised method

| | | Autoencoder | | PCA | |
|---|---|---|---|---|---|
| | Number of Features Used | Video Level | Frame Level | Video Level | Frame Level |
| No Anomalies | 4 Features | 0.72 | 0.63 | 0.72 | 0.56 |
| | 8 Features | 0.71 | 0.63 | 0.72 | 0.64 |
| | 16 Features | 0.68 | 0.55 | 0.69 | 0.55 |
| 1% Anomaly | 4 Features | 0.70 | 0.63 | 0.73 | 0.54 |
| | 8 Features | 0.71 | 0.53 | 0.72 | 0.65 |
| | 16 Features | 0.70 | 0.61 | 0.71 | 0.58 |
| 10% Anomaly | 4 Features | 0.70 | 0.60 | 0.70 | 0.51 |
| | 8 Features | 0.68 | 0.54 | 0.71 | 0.64 |
| | 16 Features | 0.68 | 0.54 | 0.72 | 0.56 |
| Entire Training Set | 4 Features | 0.68 | 0.62 | 0.66 | 0.52 |
| | 8 Features | 0.67 | 0.53 | 0.69 | 0.63 |
| | 16 Features | 0.67 | 0.53 | 0.62 | 0.70 |

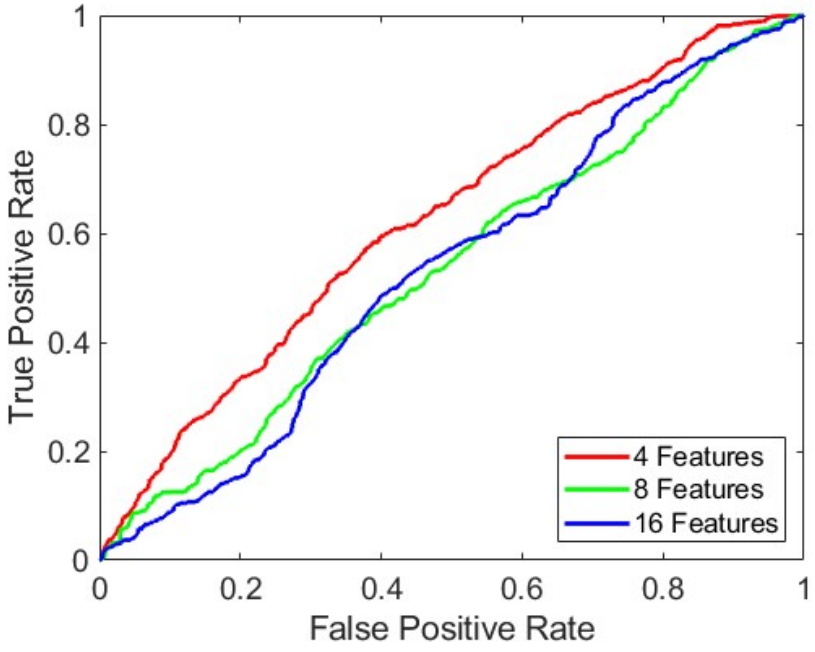



**Figure 4.3** Frame level ROC curve of AE model using the entire training set.

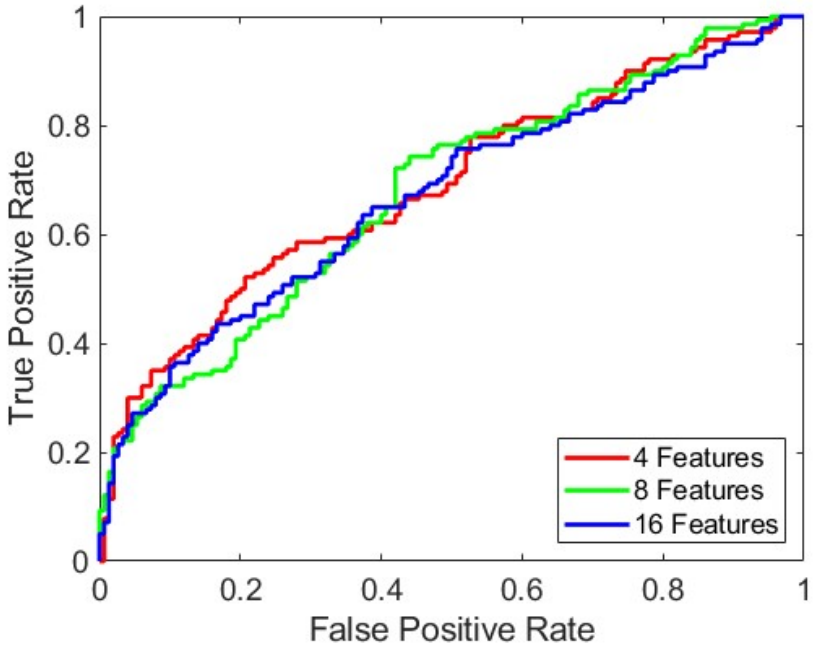



**Figure 4.4** Video level ROC curve of AE model using the entire training set.

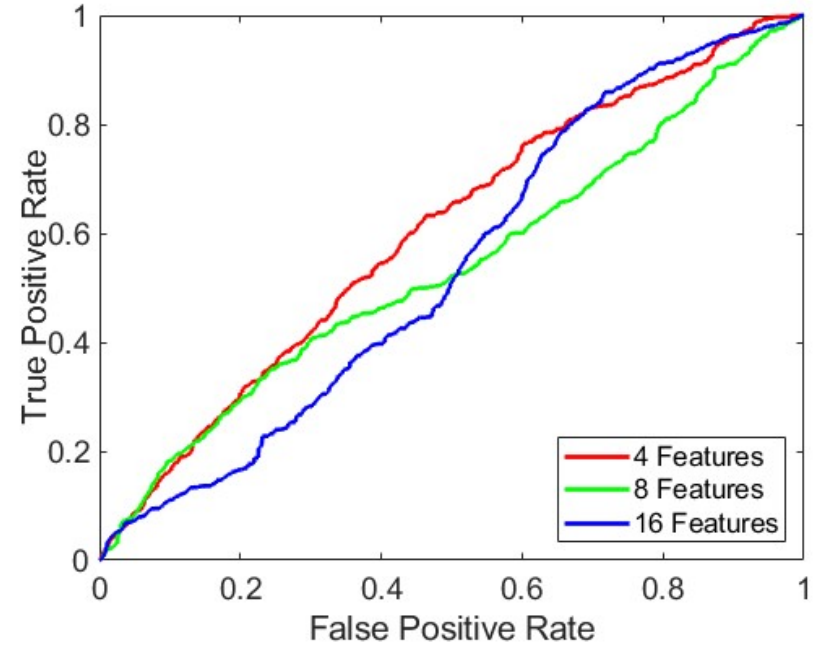



**Figure 4.5** Frame level ROC curve of AE model using normal videos and 10% of anomalous videos.

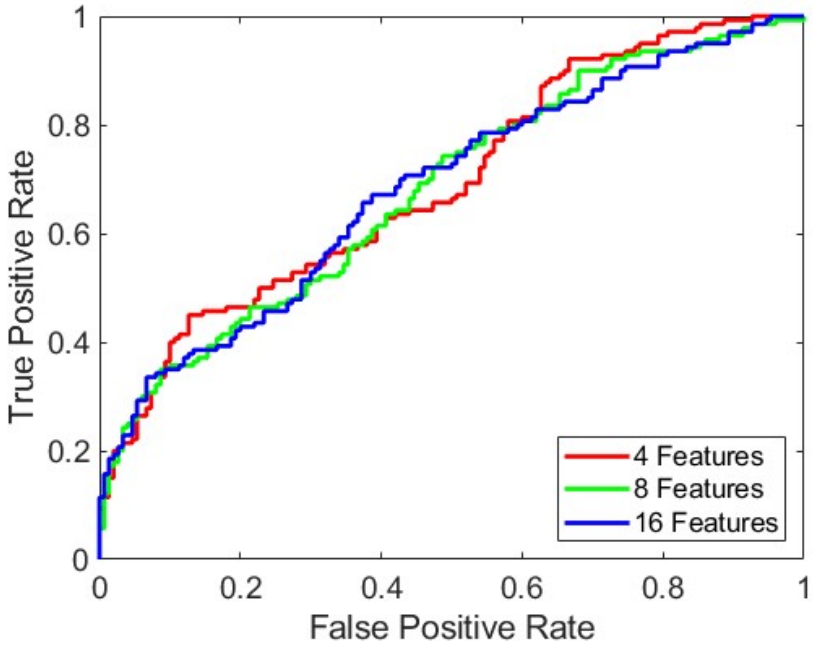



**Figure 4.6** Video level ROC curve of AE model using normal videos and 10% of anomalous videos.

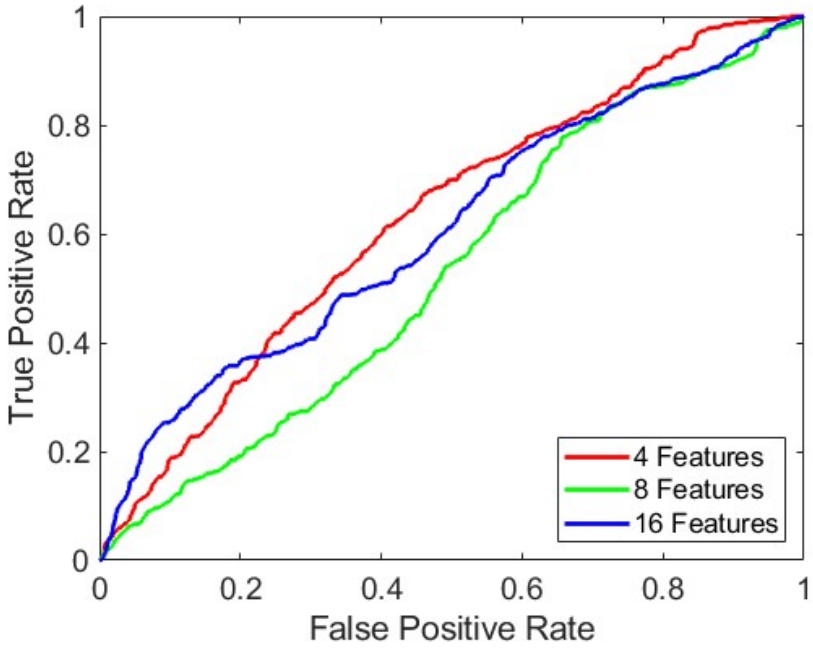



**Figure 4.7** Frame level ROC curve of AE model using normal videos and 1% of anomalous videos.

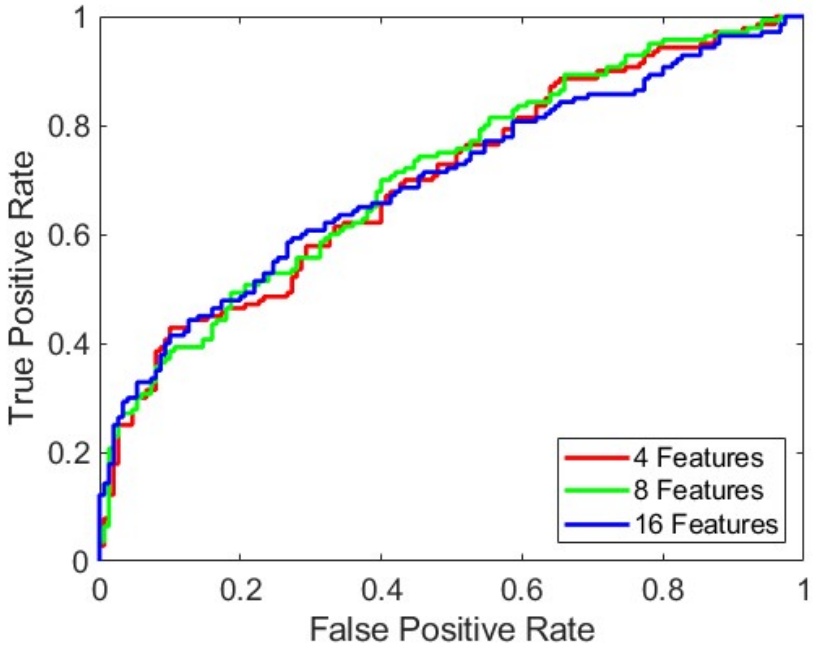



**Figure 4.8** Video level ROC curve of AE model using normal videos and 1% of anomalous videos.

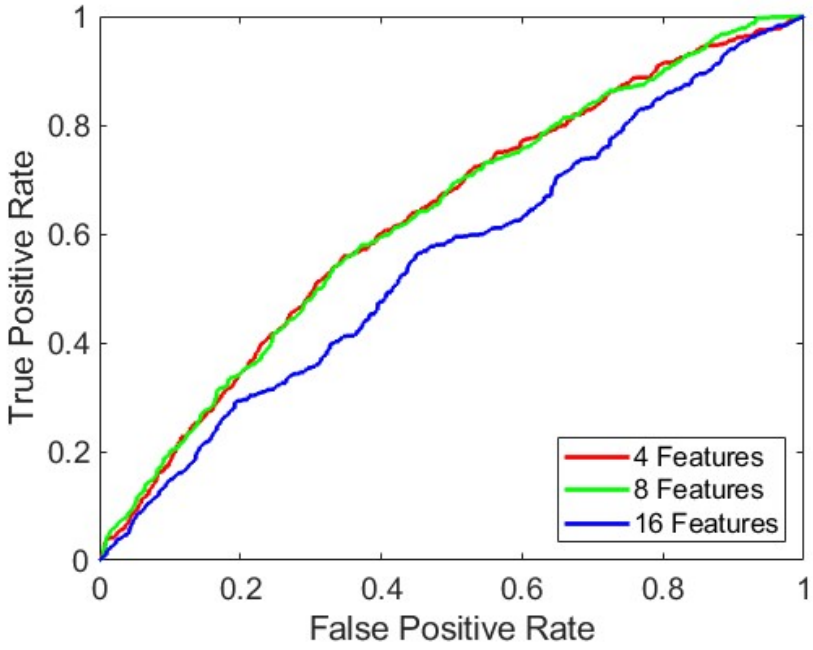



**Figure 4.9** Frame level ROC curve of AE model using only normal videos.

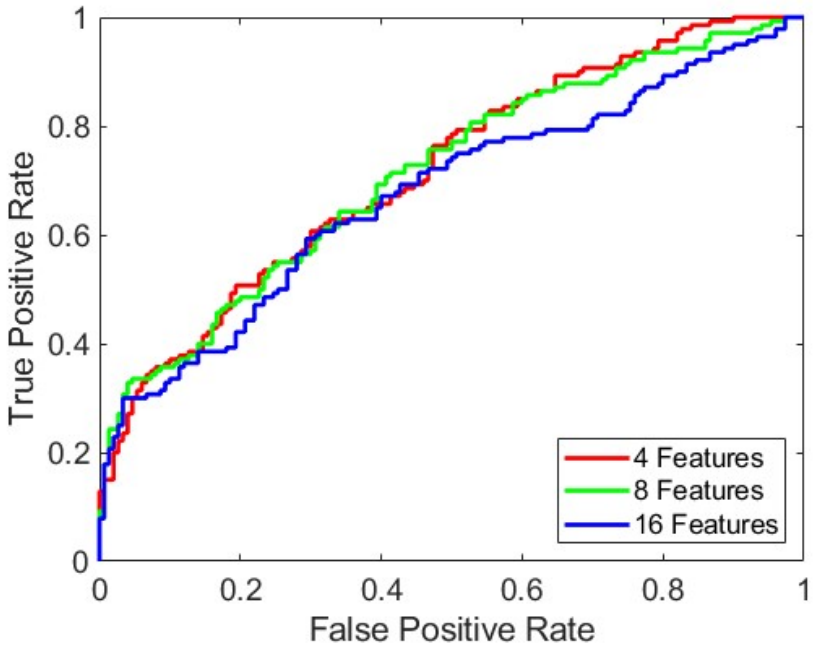



**Figure 4.10** Video level ROC curve of AE model using only normal videos.

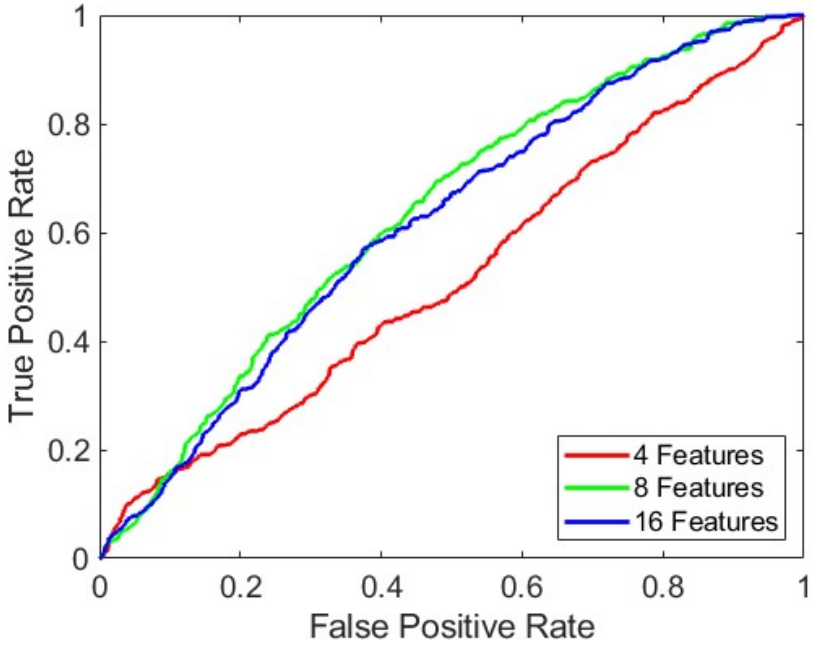



**Figure 4.11** Frame level ROC curve of PCA model using the entire training set.

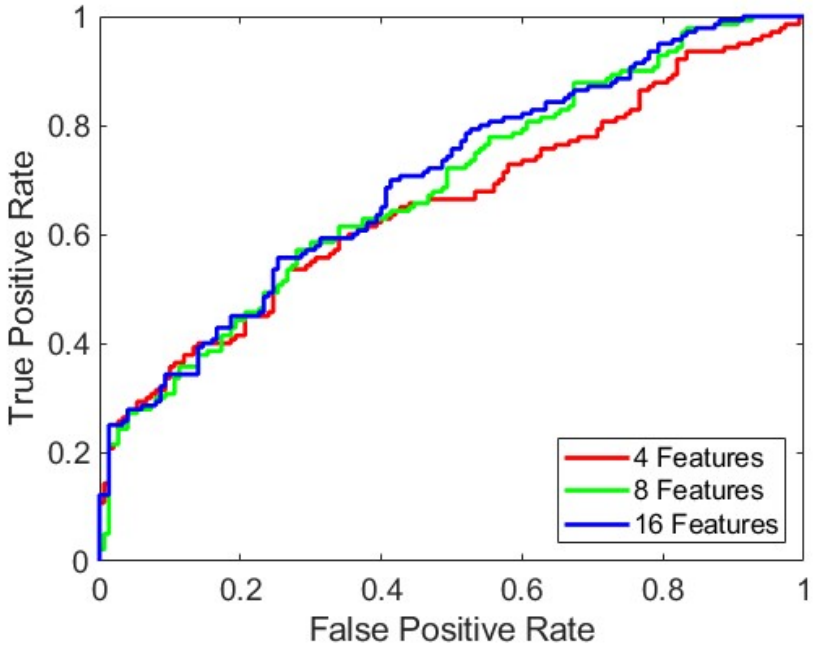



**Figure 4.12** Video level ROC curve of PCA model using the entire training set.

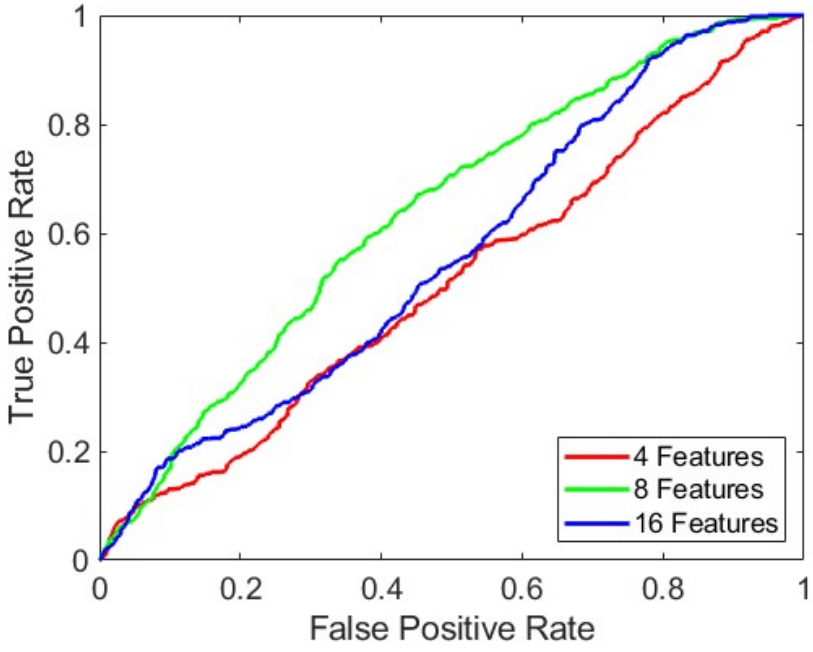



**Figure 4.13** Frame level ROC curve of PCA model using normal videos and 10% of anomalous videos.

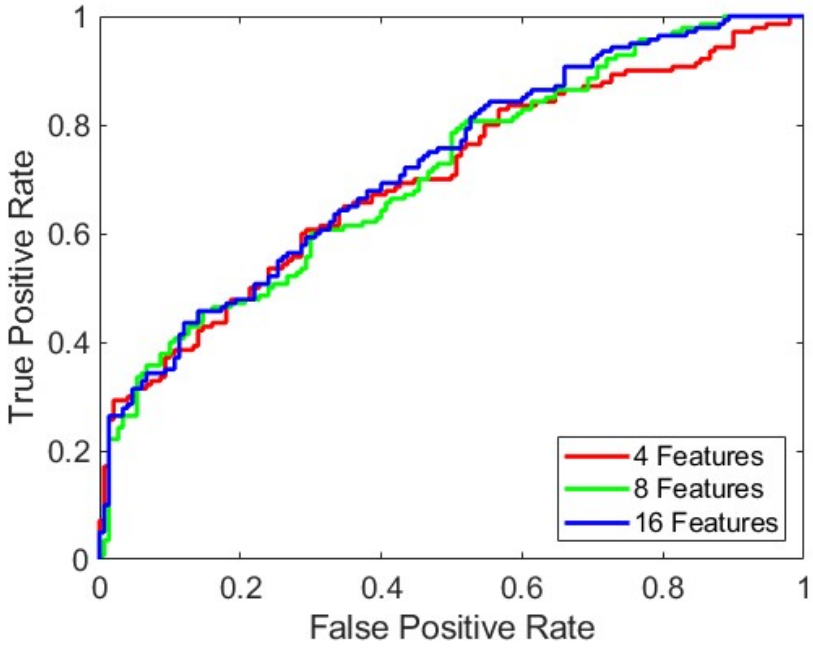



**Figure 4.14** Video level ROC curve of PCA model using normal videos and 10% of anomalous videos.

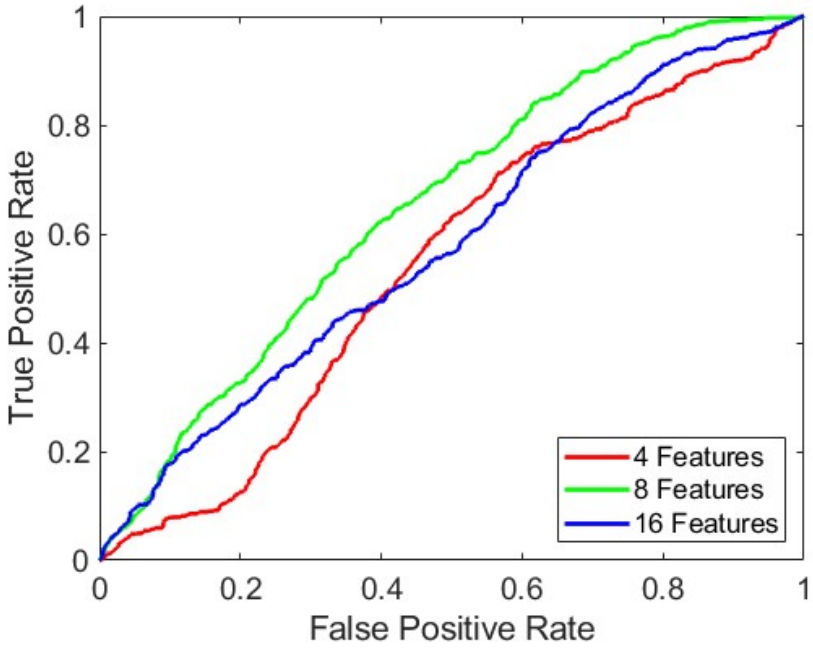



**Figure 4.15** Frame level ROC curve of PCA model using normal videos and 1% of anomalous videos.

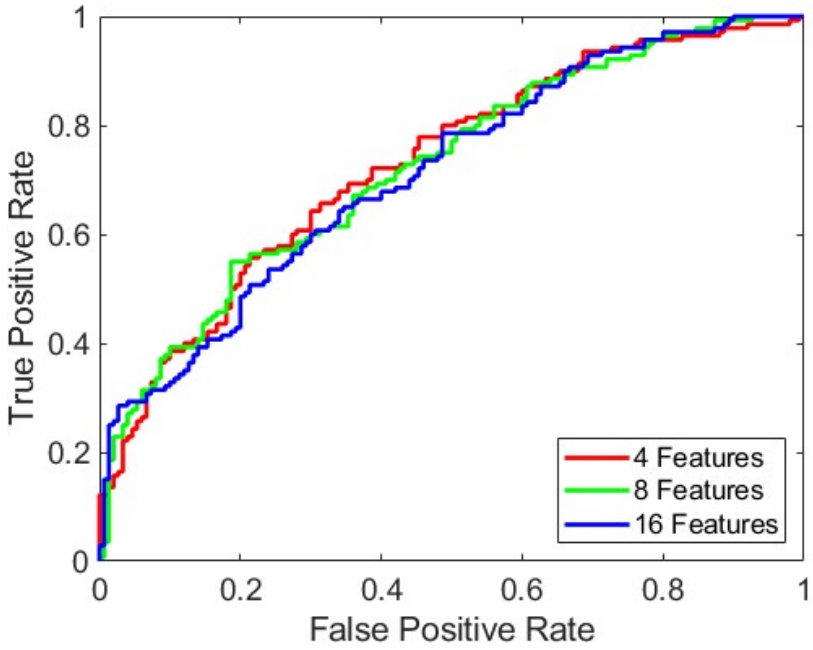



**Figure 4.16** Video level ROC curve of PCA model using normal videos and 1% of anomalous videos.

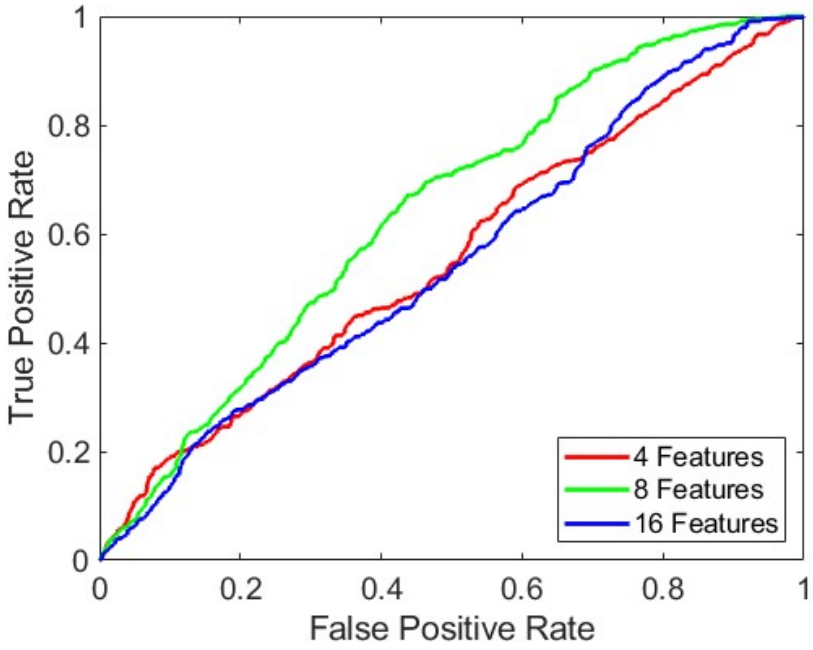



**Figure 4.17** Frame level ROC curve of PCA model using only normal videos.

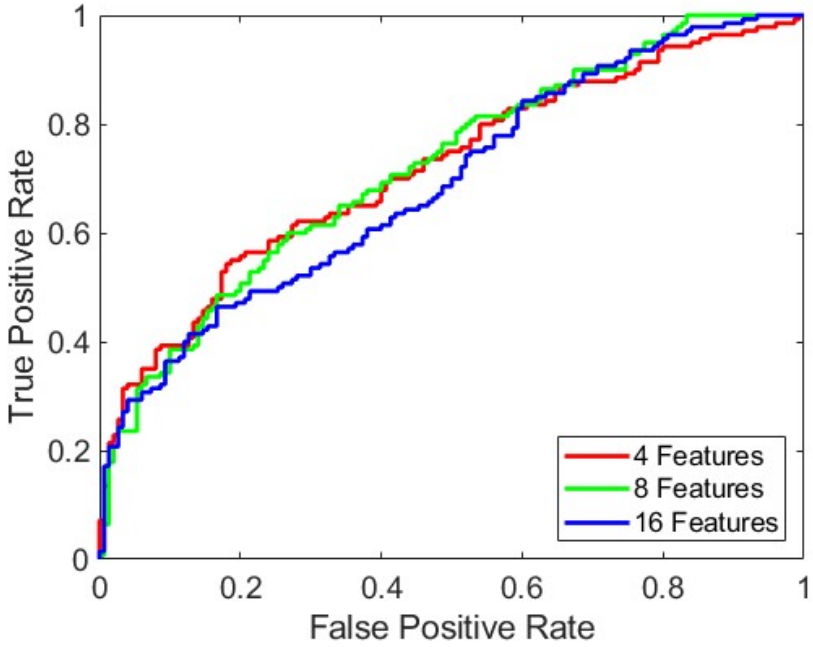



**Figure 4.18** Video level ROC curve of PCA model using only normal videos.

training partitions defined in Chapter 3.1.2. ROC curves are given in Figures 4.3-4.18 and are under the ROC curves can be seen in Table 4.3. AE model trained with no anomalies and %1 of anomalies, performed better than the rest of the models and AE works better with 4 features and PCA model works better with 8 features. We can say that even if feature extraction is unsupervised, the anomalous nature of the videos can be captured with a few features.

## 4.2 Results with the Introduced Dataset

In this section, we present our P300-based BCI speller performance results with the introduced dataset to lay out a baseline for benchmarking in later studies. For this purpose, we apply three prominent decoding algorithms (simply “decoders" or “multi-class classifiers") from the literature all of which essentially classify in test time a given EEG signal as one of the available characters. Two of these classifiers are Ensemble Support Vector Machines(ESVM) (Rakotomamonjy, Guigue, Mallet & Alvarado, 2005) and ensemble of convolutional neural networks (EoCNN) (Shan et al., 2019) as they are shown to yield a better spelling performance compared to the other popular Support Vector Machine (SVM) and convolutional neural network (CNN) based approaches. We also consider the linear discriminant analysis

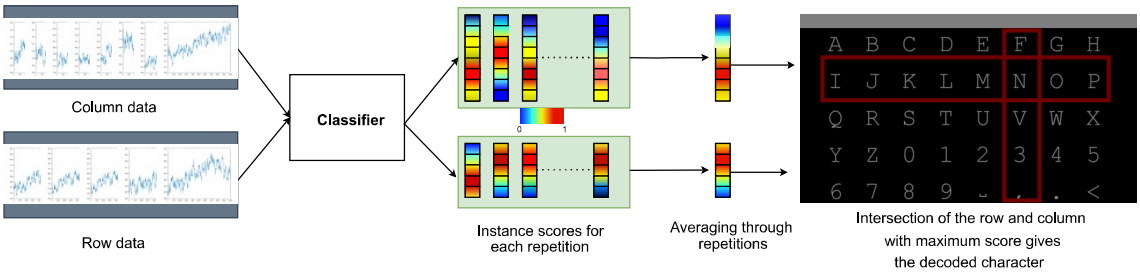



**Figure 4.19** The general decoding framework for all the methods we test is illustrated. First, the binary classification method (P300 detector) is applied to the preprocessed EEG signal from each instance of flashing for a character. Scores obtained from the method are grouped according to their row/column and repetition number, and then averaged through repetitions. Afterwards, the intersection of the row and the column with the highest score is predicted for the target character. In this figure, the high scores are represented by red color and low scores are represented with blue.

(LDA) which is another widely used classifier (Philip & George, 2020). As it is generally used in the literature, before applying these classifiers, we also firstly conduct preprocessing on the EEG signals in the introduced dataset. This preprocessing consists of linear detrending, segmentation for each flash for the first second from the onset, Fast Fourier Transform (FFT) filtering with cutoff frequencies 0.5 Hz and 30 Hz, and common average re-referencing. For the details of these steps, we refer the interested reader to (Alhaddad, 2012).

In order to visualize the P300 potential, we average the preprocessed EEG signals that are known to include (based on the labels) a P300 response, where the averaging is across all channels for each subject separately. Same process is also performed for the signals that do not include the P300. In addition to P300 plots brain activity heat maps are formed with the weights of the 32x1 feature maps from Fig. 4.20. These visualizations can be seen in Fig. 4.22.

We emphasize that although the P300-based BCI speller decoders, particularly in the RC paradigm, can be seen in general as multi-class classifiers, such decoders are typically designed based on a binary classifier that detects the presence of a P300 response in a given EEG signal. The idea is straightforward. Given a set of signals corresponding to a complete sequence of row/column flashes (in our case, we have 13 flashes in a repetition as we have 5 rows and 8 columns), one applies a binary classifier (P300 detector, where 1: P300 is present, $-1$ : P300 is absent) to each signal so that the row index and the column index of the two P300 responses can be obtained. Then, the target character is predicted as the one of the intersection of detected rows and columns. Unfortunately, due to several reasons such as the low SNR (signal-to-noise ratio) nature of EEG signals or the imperfections of the P300 detection or

perhaps the subject being inattentive, the decoding is surely prone to errors. Hence, a careful experimentation and high performance P300 detection are required for a satisfactory decoding accuracy. In this study, we specifically concentrate on three aforementioned prominent techniques that have been previously reported to deliver decent performance. Next, we explain the details of our P300 decoder performance evaluations with the introduced dataset and those techniques.

We perform the aforementioned P300-based BCI speller decoders by using the same process in each case. Namely, a 640 ms long signal is extracted for each and every row or column flash from the onset and it is downsampled to 166 samples. Here, the onset is defined as the moment the flash is given. First 5 blocks of our dataset containing 80 characters are used for training and all signals from this set are used to form the training data matrix $X_T$, where each column is a signal of length $166 \times 32$ (or an instance to be classified) since we concatenate 32 available channels. Note that we have a total of $15600 = 80$ (number of target characters in the first 5 blocks) $\times 13$ (number of flashes for each character in a repetition) $\times 15$ (number of repetitions) instances in the training set. For the training set, we also define the column vector $Y_T$ of the corresponding binary labels (where "1" is used for representing the presence for P300 response and "$-1$" for the absence). The set $X$ of the signals of all of the flashes in the remaining last 5 blocks is used for testing. For the $q$'th target character in the $i$'th block of the test set, let $X_{q,i,j,k}$ denote the corresponding instance (the column vector of concatenation of length-166 signals from all 32 channels), where $j$ indicates the number containing row-column information (1 to 8 is for columns and 9 to 13 is for rows) of the responsible flash and $k$ indicates the repetition ID (from 1 to $n \leq 15$). After this training and test split, a binary classifier (P300 detector) can be trained on the labeled training set $(X_T, Y_T)$ and applied to the test set to find the row and column scores $S^{(r)}_{q,i,j,k}$ and $S^{(c)}_{q,i,j,k}$, respectively, for each test instance. Also, $S^{(c)}_{q,i,j}$ (or $S^{(r)}_{q,i,j}$) denotes the score belonging to $q$'th target character in the $i$'th block for the $j$'th column (or row), and found by averaging $S^{(c)}_{q,i,j,k}$ (or $S^{(r)}_{q,i,j,k}$) through repetitions $k$. These scores can be interpreted as the likelihood of the rows and columns of including the target character. If $C_{q,i}$ (and $R_{q,i}$) is let represent the column (and row) with the highest score, i.e.,

$$C_{q,i} =_j (S^{(c)}_{q,i,j}) \qquad \text{FOR } j = 1, 2, ..8, \tag{4.3}$$

$$R_{q,i} =_j (S^{(r)}_{q,i,j}) \qquad \text{FOR } i = 9, 10, ..13, \tag{4.4}$$

then the character at the intersection of the $C_{q,i}$ and $R_{q,i}$ is declared as the decoded (i.e. predicted) character for the $r$'th target character in the $i$'th block. For instance, if $C_{q,i} = 6$ and $R_{q,i} = 2$, then we predict the target character as "N" since it is at the second row and sixth column. If "N" is equal to the $r$'th target character in the $i$'th block, then the decoding is accurate; and it is an error, otherwise. This process is illustrated in Fig. 4.19.

As for the performance evaluation, we use the decoding (i.e. multi-class classification) accuracy out of 80 target characters in our test set, as well as the resulting information transfer rate (ITR) which is the average number of bits that can be transferred through the BCI. The ITR is defined as (Speier, Arnold & Pouratian, 2013):

$$\text{ITR} = \frac{1}{T}\left[\log N + P\log P + (1-P)\log\frac{(1-P)}{(N-1)}\right], \tag{4.5}$$

where $P$ is the decoding accuracy, $N$ is the number of possible characters (40 in our case) and $T$ is the amount of time spelled used in the signal acquisition. Here, $T$ linearly scales with the number of repetitions used in the process (which is 15 at most in our case). Note that as $T$ increases, the decoding accuracy is expected to increase as well but then the ITR decreases.

We next provide brief information about the specific decoders ESVM, EoCNN and LDA we utilize. The goal of these decoders are essentially the same and it is to produce the row/column scores $S^{(c)}_{q,i,j}/S^{(r)}_{q,i,j}$. Once these scores are generated by using the binary classifiers of P300 detection, then the actual decoding straightforwardly follows as described above (and summarized in Fig. 4.19). The difference among these decoders underlies the P300 detectors they are based on. For the first classifier ESVM, the training set $X_T$ is further split into 16 pieces such that each piece includes consecutive characters in time. Here, the purpose of this further splitting is to capture potential non-stationarities in time. Then, a separate linear SVM is trained on each piece of training to yield a hyperplane separating P300 signals from the non-P300 signals. Namely, it yields $f_l(x) = w_l'x$, where[1] $w_l$ is the normal column vector to the hyperplane (together with the bias coefficient located at the end of the normal vector) of the $l$'th linear SVM and $x$ is an instance from the $l$'th piece. The sign of $f_l$ can be used as a P300 detector (binary classifier) or its continuous value can be used as the corresponding P300 classification score. Then, in the test phase, for each linear SVM, we obtain the column scores $S^{(c,l)}_{q,i,j,k}$ (or the row scores $S^{(r,l)}_{q,i,j,k}$) of a test instance $X_{q,i,j,k}$ as $S^{(c,l)}_{q,i,j,k} = f_l(X_{q,i,j,k}) = w_l'X_{q,i,j,k}$ if $X_{q,i,j,k}$

[1]We use $'$ to denote the transpose operation. If $X$ is a matrix, then $X'$ is its transpose.

| | | Number of Repetition | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Subject | Method | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 1 | EoCNN | 23.75 | 51.25 | 63.75 | 73.75 | 81.25 | 92.50 | 95.00 | 97.50 | 96.25 | 95.00 | 96.25 | 98.75 | 98.75 | 98.75 | 100.00 |
| | ESVM | 17.50 | 28.75 | 41.25 | 62.50 | 73.75 | 82.50 | 82.50 | 91.25 | 91.25 | 95.00 | 93.75 | 97.50 | 96.25 | 97.50 | 100.00 |
| | LDA | 1.25 | 16.25 | 37.50 | 55.00 | 68.75 | 87.50 | 90.00 | 92.50 | 98.75 | 96.25 | 96.25 | 96.25 | 98.75 | 98.75 | 100.00 |
| 2 | EoCNN | 15.00 | 38.75 | 42.50 | 53.75 | 55.00 | 65.00 | 68.75 | 70.00 | 68.75 | 71.25 | 73.75 | 76.25 | 78.75 | 78.75 | 82.50 |
| | ESVM | 16.25 | 37.50 | 48.75 | 55.00 | 57.50 | 63.75 | 66.25 | 72.50 | 71.25 | 72.50 | 72.50 | 76.25 | 81.25 | 82.50 | 86.25 |
| | LDA | 26.25 | 50.00 | 60.00 | 65.00 | 66.25 | 70.00 | 71.25 | 72.50 | 73.75 | 72.50 | 76.25 | 77.50 | 80.00 | 83.75 | 86.25 |
| 3 | EoCNN | 55.00 | 76.25 | 87.50 | 93.75 | 93.75 | 96.25 | 95.00 | 96.25 | 96.25 | 96.25 | 96.25 | 96.25 | 96.25 | 96.25 | 96.25 |
| | ESVM | 42.50 | 62.50 | 76.25 | 90.00 | 91.25 | 92.50 | 92.50 | 95.00 | 96.25 | 96.25 | 96.25 | 96.25 | 96.25 | 95.00 | 96.25 |
| | LDA | 61.25 | 77.50 | 87.50 | 91.25 | 91.25 | 93.75 | 93.75 | 96.25 | 96.25 | 96.25 | 96.25 | 96.25 | 96.25 | 96.25 | 96.25 |
| 4 | EoCNN | 6.25 | 21.25 | 30.00 | 43.75 | 55.00 | 60.00 | 72.50 | 70.00 | 77.50 | 87.50 | 87.50 | 92.50 | 93.75 | 95.00 | 96.25 |
| | ESVM | 7.50 | 16.25 | 23.75 | 28.75 | 36.25 | 45.00 | 56.25 | 62.50 | 65.00 | 73.75 | 75.00 | 68.75 | 77.50 | 78.75 | 85.00 |
| | LDA | 11.25 | 26.25 | 28.75 | 41.25 | 52.50 | 58.75 | 70.00 | 71.25 | 71.25 | 78.75 | 82.50 | 83.75 | 87.50 | 90.00 | 88.75 |
| 5 | EoCNN | 47.37 | 71.05 | 88.16 | 88.16 | 92.11 | 96.05 | 96.05 | 96.05 | 97.37 | 98.68 | 98.68 | 98.68 | 98.68 | 98.68 | 98.68 |
| | ESVM | 27.63 | 43.42 | 67.11 | 80.26 | 86.84 | 90.79 | 90.79 | 92.11 | 94.74 | 96.05 | 97.37 | 96.05 | 100.00 | 97.37 | 98.68 |
| | LDA | 19.74 | 53.95 | 80.26 | 81.58 | 92.11 | 96.05 | 93.42 | 94.74 | 96.05 | 98.68 | 98.68 | 98.68 | 97.37 | 98.68 | 98.68 |
| 6 | EoCNN | 16.25 | 35.00 | 38.75 | 58.75 | 66.25 | 71.25 | 73.75 | 75.00 | 75.00 | 80.00 | 81.25 | 83.75 | 83.75 | 85.00 | 82.50 |
| | ESVM | 11.25 | 23.75 | 35.00 | 48.75 | 60.00 | 60.00 | 66.25 | 72.50 | 73.75 | 76.25 | 76.25 | 77.50 | 77.50 | 81.25 | 78.75 |
| | LDA | 16.25 | 36.25 | 47.50 | 61.25 | 67.50 | 70.00 | 72.50 | 71.25 | 78.75 | 76.25 | 76.25 | 82.50 | 83.75 | 88.75 | 85.00 |
| 7 | EoCNN | 7.50 | 17.50 | 33.75 | 38.75 | 47.50 | 60.00 | 67.50 | 73.75 | 77.50 | 81.25 | 78.75 | 85.00 | 83.75 | 86.25 | 86.25 |
| | ESVM | 10.00 | 6.25 | 11.25 | 17.50 | 26.25 | 36.25 | 42.50 | 48.75 | 53.75 | 62.50 | 66.25 | 71.25 | 72.50 | 80.00 | 82.50 |
| | LDA | 6.25 | 22.50 | 28.75 | 41.25 | 52.50 | 63.75 | 65.00 | 70.00 | 76.25 | 77.50 | 80.00 | 82.50 | 83.75 | 85.00 | 87.50 |
| 8 | EoCNN | 11.25 | 16.25 | 10.00 | 23.75 | 42.50 | 35.00 | 31.25 | 51.25 | 51.25 | 56.25 | 56.25 | 62.50 | 56.25 | 62.50 | 66.25 |
| | ESVM | 3.75 | 11.25 | 15.00 | 21.25 | 31.25 | 40.00 | 40.00 | 43.75 | 47.50 | 52.50 | 52.50 | 61.25 | 63.75 | 57.50 | 57.50 |
| | LDA | 5.00 | 13.75 | 15.00 | 21.25 | 31.25 | 33.75 | 36.25 | 43.75 | 53.75 | 57.50 | 60.00 | 63.75 | 65.00 | 61.25 | 68.75 |
| 9 | EoCNN | 15.00 | 40.00 | 51.25 | 67.50 | 67.50 | 70.00 | 80.00 | 78.75 | 90.00 | 88.75 | 90.00 | 92.50 | 91.25 | 93.75 | 95.00 |
| | ESVM | 15.00 | 31.25 | 40.00 | 53.75 | 60.00 | 61.25 | 63.75 | 67.50 | 73.75 | 78.75 | 83.75 | 85.00 | 86.25 | 88.75 | 91.25 |
| | LDA | 1.25 | 16.25 | 48.75 | 56.25 | 65.00 | 76.25 | 83.75 | 86.25 | 85.00 | 91.25 | 93.75 | 91.25 | 95.00 | 95.00 | 95.00 |
| 10 | EoCNN | 10.00 | 17.50 | 22.50 | 35.00 | 50.00 | 56.25 | 63.75 | 62.50 | 65.00 | 71.25 | 73.75 | 81.25 | 81.25 | 87.50 | 85.00 |
| | ESVM | 5.00 | 16.25 | 13.75 | 21.25 | 26.25 | 40.00 | 46.25 | 47.50 | 50.00 | 57.50 | 61.25 | 71.25 | 70.00 | 75.00 | 71.25 |
| | LDA | 0.00 | 2.50 | 15.00 | 25.00 | 35.00 | 43.75 | 53.75 | 51.25 | 52.50 | 60.00 | 68.75 | 70.00 | 73.75 | 80.00 | 77.50 |
| 11 | EoCNN | 10.00 | 10.00 | 20.00 | 27.50 | 35.00 | 33.75 | 43.75 | 52.50 | 57.50 | 62.50 | 70.00 | 72.50 | 75.00 | 75.00 | 72.50 |
| | ESVM | 5.00 | 8.75 | 10.00 | 8.75 | 16.25 | 27.50 | 35.00 | 40.00 | 43.75 | 46.25 | 48.75 | 51.25 | 58.75 | 56.25 | 66.25 |
| | LDA | 5.00 | 2.50 | 5.00 | 16.25 | 25.00 | 32.50 | 33.75 | 45.00 | 50.00 | 56.25 | 55.00 | 60.00 | 68.75 | 65.00 | 66.25 |
| 12 | EoCNN | 21.25 | 45.00 | 52.50 | 70.00 | 81.25 | 86.25 | 92.50 | 91.25 | 95.00 | 97.50 | 97.50 | 97.50 | 96.25 | 97.50 | 97.50 |
| | ESVM | 23.75 | 36.25 | 46.25 | 70.00 | 81.25 | 86.25 | 86.25 | 86.25 | 88.75 | 93.75 | 90.00 | 90.00 | 92.50 | 95.00 | 97.50 |
| | LDA | 1.25 | 30.00 | 53.75 | 66.25 | 80.00 | 82.50 | 87.50 | 91.25 | 93.75 | 95.00 | 96.25 | 96.25 | 95.00 | 97.50 | 97.50 |
| 13 | EoCNN | 20.00 | 33.75 | 53.75 | 73.75 | 77.50 | 87.50 | 88.75 | 92.50 | 96.25 | 95.00 | 95.00 | 95.00 | 95.00 | 96.25 | 96.25 |
| | ESVM | 17.50 | 18.75 | 46.25 | 56.25 | 58.75 | 76.25 | 78.75 | 87.50 | 88.75 | 92.50 | 92.50 | 93.75 | 93.75 | 93.75 | 93.75 |
| | LDA | 1.25 | 15.00 | 36.25 | 58.75 | 72.50 | 86.25 | 87.50 | 93.75 | 96.25 | 95.00 | 95.00 | 95.00 | 95.00 | 95.00 | 95.00 |
| 14 | EoCNN | 12.50 | 23.75 | 41.25 | 60.00 | 63.75 | 76.25 | 78.75 | 78.75 | 87.50 | 88.75 | 92.50 | 93.75 | 91.25 | 91.25 | 91.25 |
| | ESVM | 11.25 | 21.25 | 25.00 | 35.00 | 48.75 | 60.00 | 67.50 | 71.25 | 75.00 | 77.50 | 83.75 | 87.50 | 87.50 | 92.50 | 91.25 |
| | LDA | 12.50 | 25.00 | 40.00 | 62.50 | 70.00 | 73.75 | 81.25 | 87.50 | 88.75 | 85.00 | 90.00 | 91.25 | 87.50 | 92.50 | 92.50 |
| 15 | EoCNN | 8.75 | 35.00 | 48.75 | 56.25 | 66.25 | 76.25 | 78.75 | 88.75 | 86.25 | 91.25 | 92.50 | 96.25 | 95.00 | 93.75 | 96.25 |
| | ESVM | 18.75 | 26.25 | 46.25 | 46.25 | 56.25 | 68.75 | 71.25 | 72.50 | 86.25 | 86.25 | 86.25 | 91.25 | 91.25 | 93.75 | 95.00 |
| | LDA | 2.50 | 25.00 | 40.00 | 53.75 | 65.00 | 73.75 | 76.25 | 77.50 | 87.50 | 88.75 | 88.75 | 93.75 | 93.75 | 96.25 | 96.25 |
| 16 | EoCNN | 4.29 | 2.86 | 5.71 | 7.14 | 7.14 | 5.71 | 5.71 | 5.71 | 2.86 | 10.00 | 15.71 | 15.71 | 21.43 | 45.71 | 58.26 |
| | ESVM | 1.43 | 12.86 | 22.86 | 34.29 | 41.43 | 44.29 | 57.14 | 62.86 | 62.86 | 74.29 | 75.71 | 84.29 | 88.57 | 87.14 | 90.00 |
| | LDA | 1.43 | 14.29 | 40.00 | 42.86 | 47.14 | 52.86 | 61.43 | 75.71 | 77.14 | 84.29 | 87.14 | 87.14 | 88.57 | 91.43 | 91.43 |
| 17 | EoCNN | 13.92 | 35.44 | 50.63 | 65.82 | 67.09 | 77.22 | 81.01 | 82.28 | 84.81 | 88.61 | 86.08 | 88.61 | 91.14 | 87.34 | 87.34 |
| | ESVM | 13.92 | 35.44 | 44.30 | 49.37 | 54.43 | 64.56 | 68.35 | 70.89 | 75.95 | 75.95 | 84.81 | 84.81 | 83.54 | 86.08 | 87.34 |
| | LDA | 13.92 | 43.04 | 56.96 | 67.09 | 69.62 | 75.95 | 78.48 | 79.75 | 82.28 | 84.81 | 88.61 | 87.34 | 89.87 | 91.14 | 92.41 |
| 18 | EoCNN | 13.75 | 35.00 | 47.50 | 58.75 | 65.00 | 73.75 | 83.75 | 86.25 | 91.25 | 92.50 | 96.25 | 96.25 | 97.50 | 96.25 | 98.75 |
| | ESVM | 18.75 | 27.50 | 37.50 | 45.00 | 58.75 | 66.25 | 73.75 | 85.00 | 83.75 | 92.50 | 95.00 | 98.75 | 98.75 | 98.75 | 97.50 |
| | LDA | 3.75 | 0.00 | 27.50 | 47.50 | 56.25 | 61.25 | 66.25 | 71.25 | 77.50 | 85.00 | 93.75 | 95.00 | 95.00 | 96.25 | 97.50 |
| Averages | EoCNN | 15.15 ± 13.51 | 29.55 ± 20.53 | 38.37 ± 24.83 | 48.09 ± 27.71 | 53.99 ± 27.88 | 59.42 ± 30.43 | 63.53 ± 30.92 | 65.61 ± 31.41 | 67.86 ± 32.77 | 70.94 ± 31.78 | 72.05 ± 31.76 | 74.37 ± 32.24 | 74.58 ± 31.48 | 76.39 ± 30.01 | 77.68 ± 28.94 |
| | ESVM | 13.00 ± 9.87 | 23.12 ± 14.69 | 31.75 ± 20.39 | 40.31 ± 24.49 | 46.80 ± 26.11 | 53.85 ± 26.35 | 57.62 ± 26.19 | 62.00 ± 26.83 | 64.57 ± 27.22 | 68.22 ± 28.64 | 69.78 ± 28.32 | 72.09 ± 28.83 | 73.97 ± 28.29 | 75.09 ± 28.54 | 76.30 ± 29.33 |
| | LDA | 9.53 ± 13.83 | 23.10 ± 19.98 | 36.36 ± 23.69 | 46.08 ± 25.10 | 53.40 ± 26.72 | 59.93 ± 27.71 | 63.37 ± 28.04 | 66.80 ± 28.30 | 69.67 ± 29.17 | 72.16 ± 28.35 | 74.02 ± 29.32 | 74.45 ± 28.76 | 76.82 ± 28.61 | 78.33 ± 28.96 | 78.75 ± 28.98 |

Table 4.4 Decoding (target identification) accuracies are given for all methods and for each subject.

comes from a row flash (or column flash). The final column score (similarly for the row score) are obtained by averaging P300 scores from all linear SVMs applied to all repetitions as

$$S^{(c)}_{q,i,j} = \frac{1}{16n} \sum_{l=1}^{16} \sum_{k=1}^{n} S^{(c,l)}_{q,i,j,k}. \tag{4.6}$$

| | | Number of Repetition | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Subject | Method | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 1 | EoCNN | 23.1208 | 40.2864 | 37.8651 | 35.8156 | 33.5511 | 34.9324 | 31.4589 | 28.9681 | 25.0924 | 22.0212 | 20.5301 | 19.842 | 18.3157 | 17.0074 | 16.3752 |
| | ESVM | 13.4979 | 15.9371 | 19.0609 | 27.5246 | 28.6524 | 28.6767 | 24.58 | 25.5657 | 22.7251 | 22.0212 | 19.5285 | 19.312 | 17.3717 | 16.5532 | 16.3752 |
| | LDA | 0.26071 | 5.888 | 16.371 | 22.5083 | 25.608 | 31.6745 | 28.5125 | 26.1993 | 26.456 | 22.5831 | 20.5301 | 18.8193 | 18.3157 | 17.0074 | 16.3752 |
| 2 | EoCNN | 10.1306 | 25.8797 | 19.9864 | 21.7095 | 18.0067 | 19.5228 | 18.2914 | 16.4712 | 14.2267 | 13.5549 | 13.0238 | 12.5992 | 12.2576 | 11.382 | 11.4707 |
| | ESVM | 11.776 | 24.5566 | 24.8248 | 22.5083 | 19.3101 | 18.9325 | 17.2464 | 17.4225 | 15.0611 | 13.938 | 12.6709 | 12.5992 | 12.9043 | 12.29 | 12.3616 |
| | LDA | 27.3899 | 38.7514 | 34.4126 | 29.2843 | 24.1449 | 21.9617 | 19.3642 | 17.4225 | 15.918 | 13.938 | 13.7446 | 12.9366 | 12.5784 | 12.6026 | 12.3616 |
| 3 | EoCNN | 90.0334 | 75.5951 | 63.3491 | 53.7035 | 42.9628 | 37.6386 | 31.4589 | 28.2289 | 25.0924 | 22.5831 | 20.5301 | 18.8193 | 17.3717 | 16.1308 | 15.0554 |
| | ESVM | 59.9591 | 55.0493 | 50.3967 | 49.8968 | 40.9051 | 34.9324 | 29.9421 | 27.5265 | 25.0924 | 22.5831 | 20.5301 | 18.8193 | 17.3717 | 15.7294 | 15.0554 |
| | LDA | 106.6462 | 77.6197 | 63.3491 | 51.1314 | 40.9051 | 35.8023 | 30.6877 | 28.2289 | 25.0924 | 22.5831 | 20.5301 | 18.8193 | 17.3717 | 16.1308 | 15.0554 |
| 4 | EoCNN | 1.3649 | 9.5409 | 11.3978 | 15.6945 | 18.0067 | 17.2063 | 19.9114 | 16.4712 | 17.2488 | 19.0047 | 17.277 | 17.4662 | 16.5241 | 15.7294 | 15.0554 |
| | ESVM | 2.244 | 5.888 | 7.7069 | 7.9686 | 9.3023 | 10.9399 | 13.3244 | 13.7623 | 13.0152 | 14.3262 | 13.3817 | 10.67 | 11.9415 | 11.382 | 12.0593 |
| | LDA | 5.7103 | 13.695 | 10.6248 | 14.2957 | 16.7369 | 16.6454 | 18.8243 | 16.9437 | 15.0611 | 15.9348 | 15.6418 | 14.7031 | 14.619 | 14.2562 | 12.9843 |
| 5 | EoCNN | 71.1755 | 67.4747 | 64.1726 | 48.1295 | 41.5957 | 37.4878 | 32.1324 | 28.1159 | 25.6783 | 23.7753 | 21.6139 | 19.8128 | 18.2887 | 16.9824 | 15.8502 |
| | ESVM | 29.8423 | 31.016 | 41.0686 | 41.1014 | 37.5206 | 33.782 | 28.956 | 25.9973 | 24.3399 | 22.4927 | 21.0095 | 18.7439 | 18.8944 | 16.5075 | 15.8502 |
| | LDA | 16.7568 | 43.6699 | 54.8019 | 42.2204 | 41.5957 | 37.4878 | 30.4892 | 27.3824 | 24.9919 | 23.7753 | 21.6139 | 19.8128 | 17.7773 | 16.9824 | 15.8502 |
| 6 | EoCNN | 11.776 | 21.9773 | 17.2531 | 24.9681 | 24.1449 | 22.5916 | 20.466 | 18.3998 | 16.3554 | 16.3519 | 15.2505 | 14.7031 | 13.572 | 12.9207 | 11.4707 |
| | ESVM | 5.7103 | 11.5604 | 14.6515 | 18.6186 | 20.6475 | 17.2063 | 17.2464 | 17.4225 | 15.918 | 15.119 | 13.7446 | 12.9366 | 11.9415 | 11.9825 | 10.6232 |
| | LDA | 11.776 | 23.2557 | 23.8292 | 26.6615 | 24.8717 | 21.9617 | 19.9114 | 16.9437 | 17.7054 | 15.119 | 13.7446 | 14.3383 | 13.572 | 13.9118 | 12.0593 |
| 7 | EoCNN | 2.244 | 6.749 | 13.8145 | 12.9398 | 14.2975 | 17.2063 | 17.7655 | 17.9078 | 17.2488 | 16.7756 | 14.4862 | 15.0741 | 13.572 | 13.2446 | 12.3616 |
| | ESVM | 4.434 | 0.68246 | 1.9034 | 3.3745 | 5.478 | 7.7519 | 8.5656 | 9.3093 | 9.6487 | 11.0099 | 10.975 | 11.2958 | 10.7215 | 11.68 | 11.4707 |
| | LDA | 1.3649 | 10.5357 | 10.6248 | 14.2957 | 16.7369 | 18.9325 | 16.7339 | 16.4712 | 16.7989 | 15.5239 | 14.8654 | 14.3383 | 13.572 | 12.9207 | 12.6698 |
| 8 | EoCNN | 5.7103 | 5.888 | 1.478 | 5.7802 | 11.9918 | 7.3258 | 5.2231 | 10.0716 | 8.9525 | 9.3271 | 8.4792 | 9.1749 | 7.1747 | 7.8642 | 8.0483 |
| | ESVM | 0.18547 | 2.8551 | 3.3769 | 4.7705 | 7.3124 | 9.0749 | 7.7785 | 7.8472 | 7.9431 | 8.3685 | 7.6077 | 8.8872 | 8.7381 | 6.8965 | 6.4367 |
| | LDA | 0.66457 | 4.2834 | 3.3769 | 4.7705 | 7.3124 | 6.9073 | 6.6445 | 7.8472 | 9.6487 | 9.655 | 9.3852 | 9.4663 | 9.0105 | 7.6176 | 8.536 |
| 9 | EoCNN | 10.1306 | 27.2247 | 26.8576 | 31.0896 | 24.8717 | 21.9617 | 23.3599 | 19.9185 | 22.1764 | 19.4765 | 18.1443 | 17.4662 | 15.7327 | 15.3438 | 14.6808 |
| | ESVM | 10.1306 | 18.281 | 18.1498 | 21.7095 | 20.6475 | 17.7744 | 16.2279 | 15.5448 | 15.918 | 15.9348 | 16.0397 | 15.0741 | 14.2634 | 13.9118 | 13.635 |
| | LDA | 0.26071 | 5.888 | 24.8248 | 23.3177 | 23.4274 | 25.1984 | 25.2052 | 23.1781 | 20.0989 | 20.4525 | 19.5285 | 17.0438 | 16.9394 | 15.7294 | 14.6808 |
| 10 | EoCNN | 4.434 | 6.749 | 7.0238 | 10.9886 | 15.5006 | 15.5451 | 16.2279 | 13.7623 | 13.0152 | 13.5549 | 13.0238 | 13.9796 | 12.9043 | 13.5748 | 12.0593 |
| | ESVM | 0.66457 | 5.888 | 2.8556 | 4.7705 | 5.478 | 9.0749 | 9.7918 | 8.9359 | 8.6114 | 9.655 | 9.6951 | 11.2958 | 10.1362 | 10.5142 | 9.0366 |
| | LDA | 0 | 0 | 3.3769 | 6.3069 | 8.7909 | 10.463 | 12.4054 | 10.0716 | 9.2983 | 10.3238 | 11.64 | 10.9808 | 11.0202 | 11.68 | 10.3493 |
| 11 | EoCNN | 4.434 | 2.217 | 5.7181 | 7.4015 | 8.7909 | 6.9073 | 8.9683 | 10.4606 | 10.7278 | 11.0099 | 11.9791 | 11.615 | 11.323 | 10.5142 | 9.292 |
| | ESVM | 0.66457 | 1.6368 | 1.478 | 0.81841 | 2.3552 | 4.9343 | 6.2792 | 6.8062 | 6.9753 | 6.8542 | 6.7704 | 6.7144 | 7.6825 | 6.6622 | 8.0483 |
| | LDA | 0.66457 | 0 | 0.22152 | 2.944 | 5.0455 | 6.4965 | 5.9205 | 8.2049 | 8.6114 | 9.3271 | 8.1849 | 8.6031 | 9.8492 | 8.3669 | 8.0483 |
| 12 | EoCNN | 19.0818 | 32.8196 | 27.8949 | 32.9425 | 33.5511 | 30.9041 | 29.9421 | 25.5657 | 24.468 | 23.1745 | 21.0677 | 19.312 | 17.3717 | 16.5532 | 15.4496 |
| | ESVM | 23.1208 | 23.2557 | 22.8475 | 32.9425 | 33.5511 | 30.9041 | 26.4892 | 23.1781 | 21.6406 | 21.4814 | 18.1443 | 16.6323 | 16.1226 | 15.7294 | 15.4496 |
| | LDA | 0.26071 | 17.0967 | 28.946 | 30.1811 | 32.7039 | 28.6767 | 27.1496 | 25.5657 | 23.8682 | 22.0212 | 20.5301 | 18.8193 | 16.9394 | 16.5532 | 15.4496 |
| 13 | EoCNN | 17.1544 | 20.7218 | 28.946 | 35.8156 | 31.0479 | 31.6745 | 27.8236 | 26.1993 | 25.0924 | 22.0212 | 20.0193 | 18.351 | 16.9394 | 16.1308 | 15.0554 |
| | ESVM | 13.4979 | 7.646 | 22.8475 | 23.3177 | 19.9745 | 25.1984 | 22.764 | 23.7559 | 21.6406 | 20.9594 | 19.054 | 17.9012 | 16.5241 | 15.3438 | 14.3209 |
| | LDA | 0.26071 | 5.0653 | 15.5038 | 24.9681 | 27.8759 | 30.9041 | 27.1496 | 26.8517 | 25.0924 | 22.0212 | 20.0193 | 18.351 | 16.9394 | 15.7294 | 14.6808 |
| 14 | EoCNN | 7.0909 | 11.5604 | 19.0609 | 25.8094 | 22.7191 | 25.1984 | 22.764 | 19.9185 | 21.1164 | 19.4765 | 19.054 | 17.9012 | 15.7327 | 14.609 | 13.635 |
| | ESVM | 5.7103 | 9.5409 | 8.4092 | 10.9886 | 14.8949 | 17.2063 | 17.7655 | 16.9437 | 16.3554 | 15.5239 | 16.0397 | 15.8373 | 14.619 | 14.971 | 13.635 |
| | LDA | 7.0909 | 12.6138 | 18.1498 | 27.5246 | 26.354 | 23.877 | 23.9651 | 23.7559 | 21.6406 | 18.089 | 18.1443 | 17.0438 | 14.619 | 14.971 | 13.973 |
| 15 | EoCNN | 3.2736 | 21.9773 | 24.8248 | 23.3177 | 24.1449 | 25.1984 | 22.764 | 24.3456 | 20.6027 | 20.4525 | 19.054 | 18.8193 | 16.9394 | 15.3438 | 15.0554 |
| | ESVM | 15.292 | 13.695 | 22.8475 | 17.1356 | 18.6541 | 21.34 | 19.3642 | 17.4225 | 20.6027 | 18.5425 | 16.8568 | 17.0438 | 15.7327 | 15.3438 | 14.6808 |
| | LDA | 0 | 12.6138 | 18.1498 | 21.7095 | 23.4274 | 23.877 | 21.5986 | 19.4049 | 21.1164 | 19.4765 | 17.7059 | 17.9012 | 16.5241 | 16.1308 | 15.0554 |
| 16 | EoCNN | 0.36005 | 0.0083333 | 0.34695 | 0.4941 | 0.39528 | 0.17347 | 0.14869 | 0.13011 | 0.0018519 | 0.4434 | 1.0056 | 0.92176 | 1.4894 | 4.8066 | 5.6388 |
| | ESVM | 0.185 | 3.7516 | 7.217 | 10.6285 | 11.5154 | 10.6665 | 13.6584 | 13.8866 | 12.3437 | 14.4942 | 13.5884 | 14.8613 | 14.9296 | 13.4798 | 13.3058 |
| | LDA | 0.185 | 4.6133 | 18.1498 | 15.19 | 14.1284 | 14.0969 | 15.3052 | 18.6841 | 17.1196 | 17.8335 | 17.1561 | 15.7264 | 14.9296 | 14.6601 | 13.6827 |
| 17 | EoCNN | 8.7795 | 22.4278 | 26.3507 | 29.8733 | 24.6315 | 25.7186 | 23.8494 | 21.4115 | 20.0231 | 19.4222 | 16.7989 | 16.1852 | 15.6987 | 13.5326 | 12.6305 |
| | ESVM | 8.7795 | 22.4278 | 21.3468 | 18.991 | 17.7144 | 19.3128 | 18.1243 | 16.8055 | 16.6917 | 16.2669 | 16.3826 | 15.0173 | 13.5163 | 13.1991 | 12.6305 |
| | LDA | 8.7795 | 30.5835 | 31.7112 | 30.7893 | 26.1263 | 25.0375 | 22.637 | 20.3337 | 19.0324 | 18.0208 | 17.6566 | 15.7881 | 15.3149 | 14.5774 | 13.947 |
| 18 | EoCNN | 8.5668 | 21.9773 | 23.8292 | 24.9681 | 23.4274 | 23.877 | 25.2052 | 23.1781 | 22.7251 | 20.9594 | 20.5301 | 18.8193 | 17.8265 | 16.1308 | 15.8736 |
| | ESVM | 8.5668 | 14.803 | 16.371 | 16.4098 | 19.9745 | 20.1208 | 20.466 | 22.6112 | 19.6041 | 20.9594 | 20.0193 | 19.842 | 18.3157 | 17.0074 | 15.4496 |
| | LDA | 0.18547 | 0 | 9.8687 | 17.8719 | 18.6541 | 17.7744 | 17.2464 | 16.9437 | 17.2488 | 18.089 | 19.5285 | 18.351 | 16.9394 | 16.1308 | 15.4496 |
| Averages | EoCNN | 16.6034 ± 24.2986 | 23.393 ± 20.6726 | 23.3427 ± 17.8859 | 24.5245 ± 14.2147 | 22.9799 ± 10.9991 | 22.2817 ± 10.5009 | 20.9867 ± 9.001 | 19.418 ± 7.5282 | 18.3247 ± 6.9101 | 17.4103 ± 5.9677 | 16.2149 ± 5.3012 | 15.6034 ± 4.7657 | 14.3908 ± 4.3322 | 13.7667 ± 3.32 | 13.0588 ± 2.9941 |
| | ESVM | 11.9034 ± 14.4521 | 14.915 ± 13.3013 | 17.0755 ± 13.2822 | 18.5264 ± 13.2629 | 18.5493 ± 11.0601 | 18.7683 ± 9.1256 | 17.6983 ± 7.0723 | 17.2635 ± 6.4803 | 16.3404 ± 5.7304 | 16.1406 ± 4.9973 | 15.1132 ± 4.3679 | 14.638 ± 3.8079 | 13.9848 ± 3.2872 | 13.288 ± 3.0666 | 12.8014 ± 2.7994 |
| | LDA | 10.4587 ± 25.1317 | 17.0096 ± 19.9197 | 21.4607 ± 16.7695 | 22.5539 ± 12.3687 | 22.4692 ± 10.1711 | 22.0986 ± 9.0548 | 20.5417 ± 7.3659 | 19.4685 ± 6.3819 | 18.6 ± 5.5669 | 17.4871 ± 4.5386 | 16.675 ± 4.007 | 15.6579 ± 3.3644 | 14.824 ± 2.7624 | 14.2199 ± 2.7218 | 13.4005 ± 2.3999 |

Table 4.5 ITR values are given for all methods and for each subject.

The classifier LDA also finds a hyperplane that separates the P300 signals from the non-P300 signals. However, the hyperplane of LDA is fundamentally different from the one of the linear SVM. In the LDA, the classes are assumed to be Gaussian distributed in which case there is a closed form solution for the normal vector $w$, whereas the linear SVM finds the hyperplane as directly a separator with geometrically the maximum margin from both classes in which case there is no closed form solution and instead a quadratic optimization is performed. On the other hand, LDA is equivalent to linear regression (up to scaling) with binary response variables ("1" or "$-1$" as in the case of our P300 classification) when the instances are zero

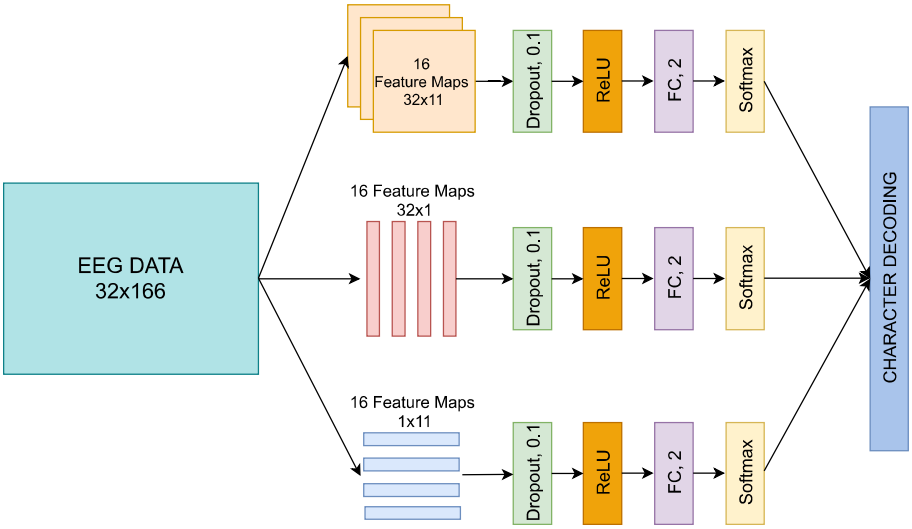



**Figure 4.20** The CNN structure of the decoder EoCNN Shan et al. (2019).

mean. Hence, the normal vector (again with the bias coefficient in the end) $w$ for LDA is obtained as (Kindermans et al., 2012)

$$w = (X_T X_T' + \alpha I)^{-1} X_T Y_T, \tag{4.7}$$

where we note that $X_T X_T'$ is the unnormalized covariance (when the instances are zero mean) and $X_T Y_T$ yields the difference between the two class means. With these substitutions, one can obtain the conventional LDA definition (with Gaussian class specific distributions of equal covariance). The regularization coefficient $\alpha \geq 0$ is used to mitigate overfitting that is typically found by cross-validation (5-fold in this study) for every subject separately. Now, in this case, the column scores (similarly for the row scores) can be obtained as

$$S_{q,i,j}^{(c)} = \frac{1}{n} \sum_{k=1}^{n} w' X_{q,i,j,k}. \tag{4.8}$$

CNN is widely used in image processing (Krizhevsky, Sutskever & Hinton, 2017) and recently started gaining attention for P300 classification tasks (Cecotti & Graser, 2011; Liu, Wu, Gu, Yu, Qi & Li, 2018) as well. Accordingly, the third and last method we consider for character decoding is EoCNN that is based on an ensemble of three separate CNNs. The network structure for this decoder is illustrated in Fig. 4.20, and we refer to the study (Shan et al., 2019) for the details. In order to apply EoCNN and make use of its constituent CNNs, we rearrange the signals by representing them in two dimension as a $32 \times 166$ matrix, where 32 is the number

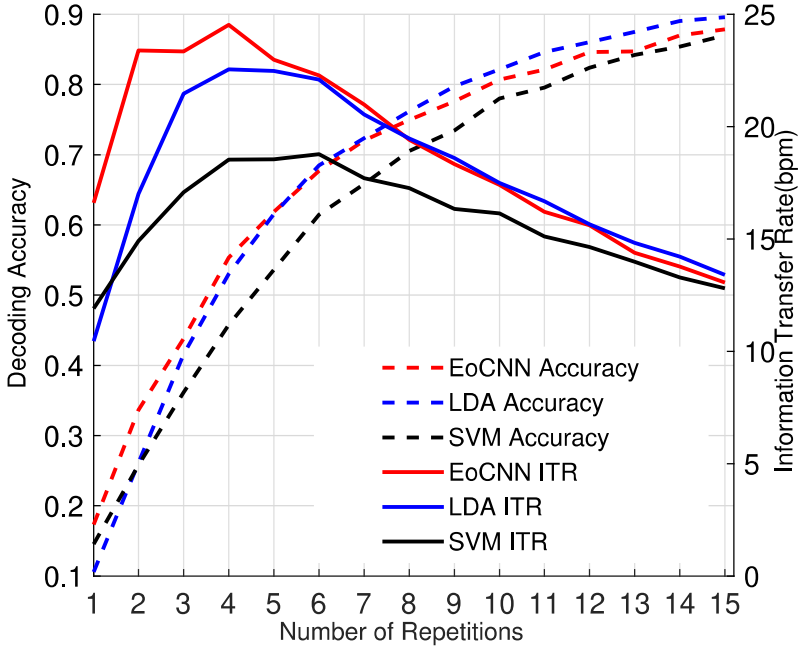



**Figure 4.21** Accuracy and ITR results (averaged across all 18 subjects) for the methods we test are plotted across various numbers of repetitions.

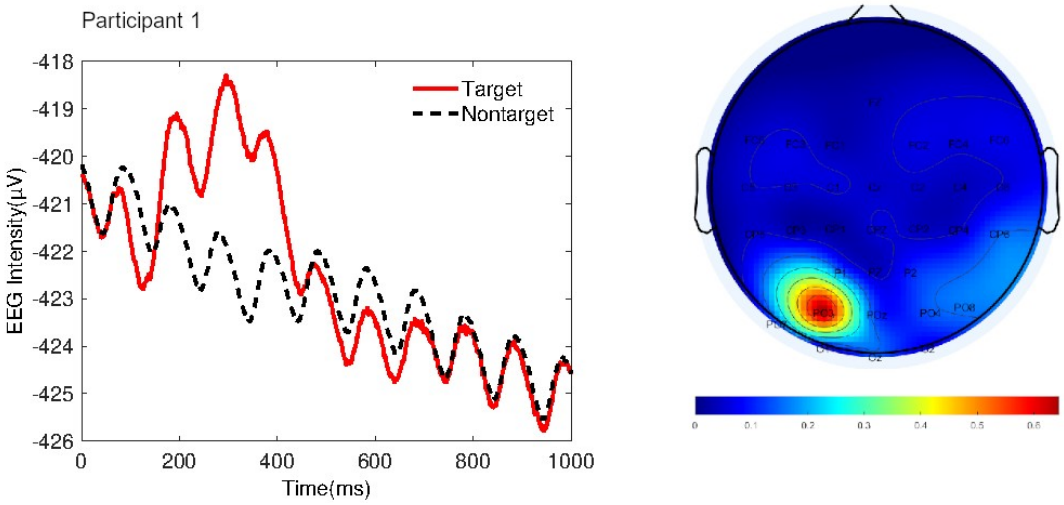
of channels. We input this matrix to each of the three CNNs, where the first CNN (denoted by $N_1$) is designed to learn the temporal relationships, the second CNN ($N_2$) learns the good channel combinations, and the third CNN ($N_3$) takes into account the temporal and spatial (channel information) relationships simultaneously. The three CNNs are trained separately and their last softmax output scores are used in decoding as described previously. Namely,

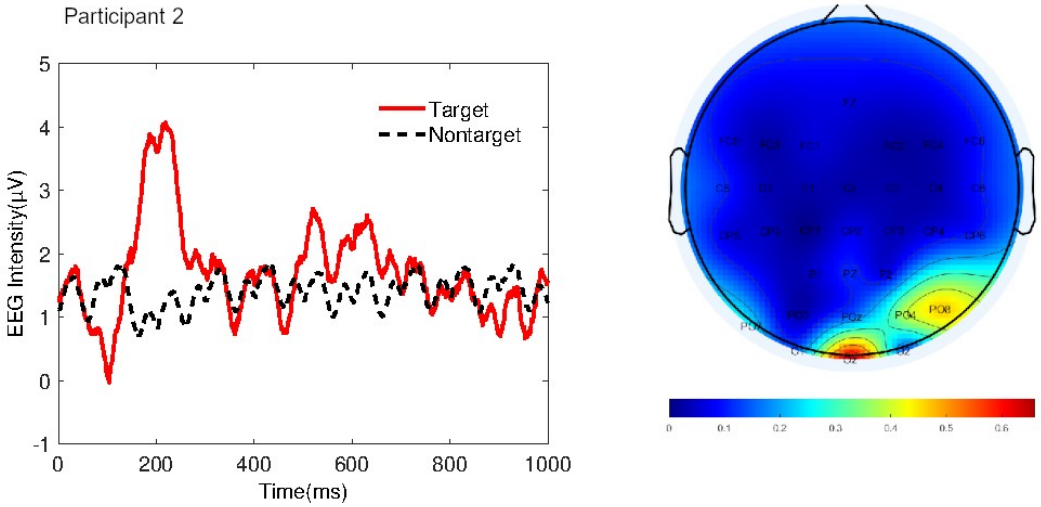
$$S^{(c)}_{q,i,j} = \frac{1}{3n} \sum_{k=1}^{n} \sum_{l=1}^{3} \mathrm{N}_l(X_{q,i,j,k}), \tag{4.9}$$

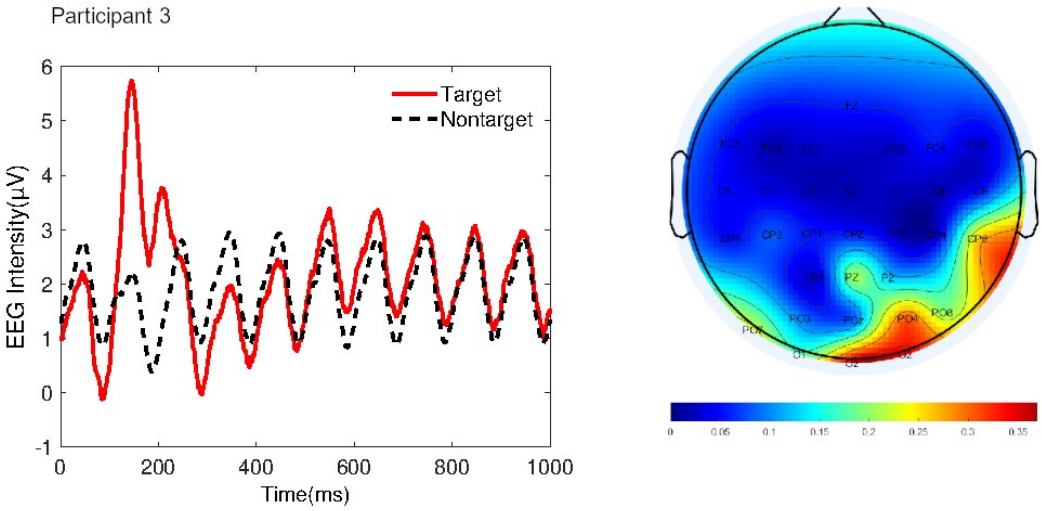
and similarly for the row scores.

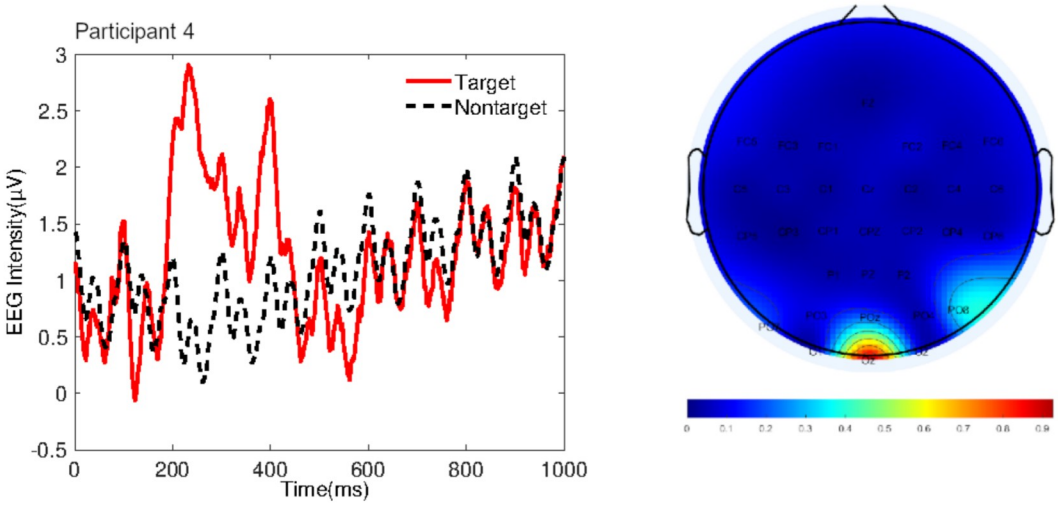
For all these techniques, we reserve 5 blocks of our introduced dataset for training and the remaining 5 blocks for testing. Regularization parameter optimization for LDA is based on 5-fold cross validation (subject-wise), whereas we use the box constraint for the SVM as 1 (linear SVM). For the network in the technique EoCNN, ReLu activation has been used with dropout layers of probability 0.1. The parameter optimization is based on the full scale of data coming from all 15 repetitions, and the same optimized parameter values have been used for data restricted to any smaller number of repetitions when checking the effect of repetition on the decoding accuracy or ITR. Further details can be found in the Supplementary document.

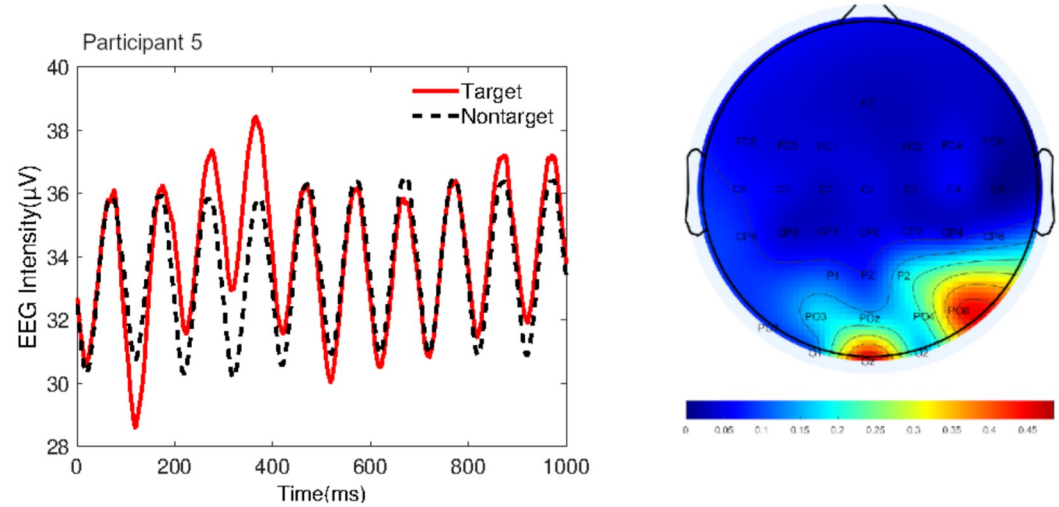
We present our accuracy and information transfer rate (ITR) results in Fig. 4.21, where all three techniques approach the level of $85-90\%$ decoding accuracy as the repetition number reaches to 15. LDA and EoCNN perform comparably, but both of them outperform SVM. The highest ITR (around 25 bpm) is observed when using 4 repetitions with the CNN based technique EoCNN. On the other hand, EoCNN has been previously reported to achieve 47.33 (higher than our report in Fig. 4.21) and 20.99 (comparable to our report in Fig. 4.21) bpm with the BCI Competition Dataset III Subject A and Subject B, respectively (Blankertz et al., 2006). We point out that these previously reported ITR figures of EoCNN are for two specific subjects, whereas the ITR performance we report in Fig. 4.21 is an average ITR across 18 subjects. The complete set of our ITR results for all 18 subjects is provided in the Supplementary document, which includes successful subjects with ITR performance as high as 90 bpm as well as poor performing subjects with ITR performance as low as 6.57 bpm. Finally, our introduced dataset has been generated in much faster speller settings compared to the BCI Competition Dataset III (SD: 66.6 ms and ISI: 33.3 ms in our case vs SD: 100 ms and ISI: 75 ms in theirs). Therefore, we certainly present a more challenging P300 detection problem in this study due to the strong P300 interference as a result of short SD and ISI in our dataset. This naturally pulls down the average ITR performance we observe here because most of the current P300 decoding methods (including the ones we use here) in the literature have been previously designed for relatively slower spellers (i.e. they are essentially sub-optimal for fast spellers such as the dataset we introduced). Thus, new P300 detection studies must be conducted to address the requirements of more realistic and relatively faster spellers.

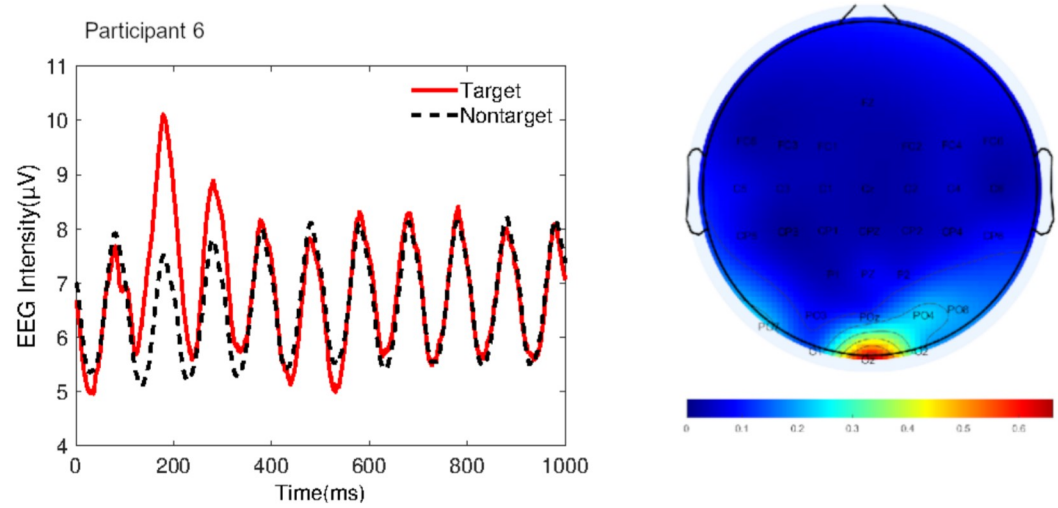

Participant 1
Target
Nontarget
EEG Intensity(μV)
Time(ms)
-418
-419
-420
-421
-422
-423
-424
-425
-426
0
200
400
600
800
1000

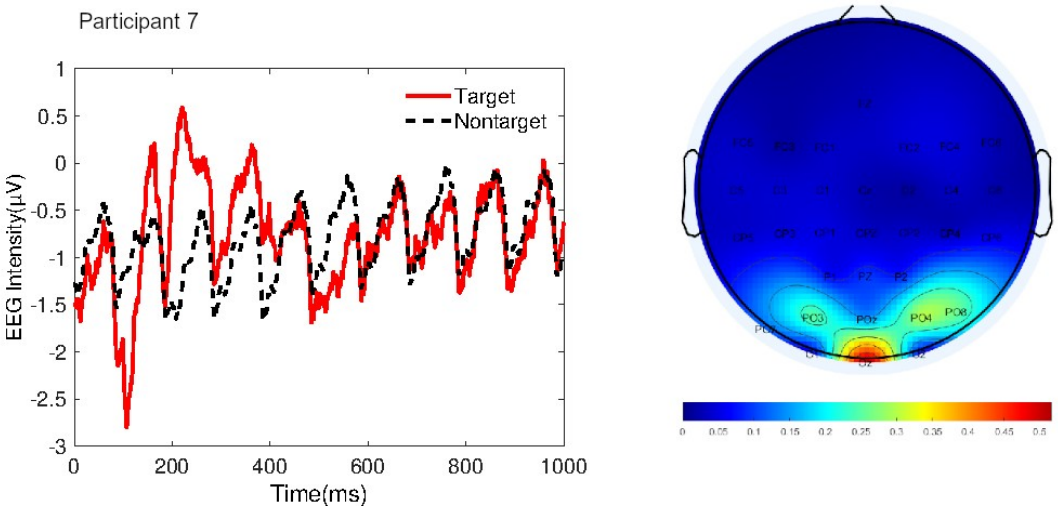

Participant 2
Target
Nontarget
EEG Intensity(μV)
Time(ms)
5
4
3
2
1
0
-1
0
200
400
600
800
1000

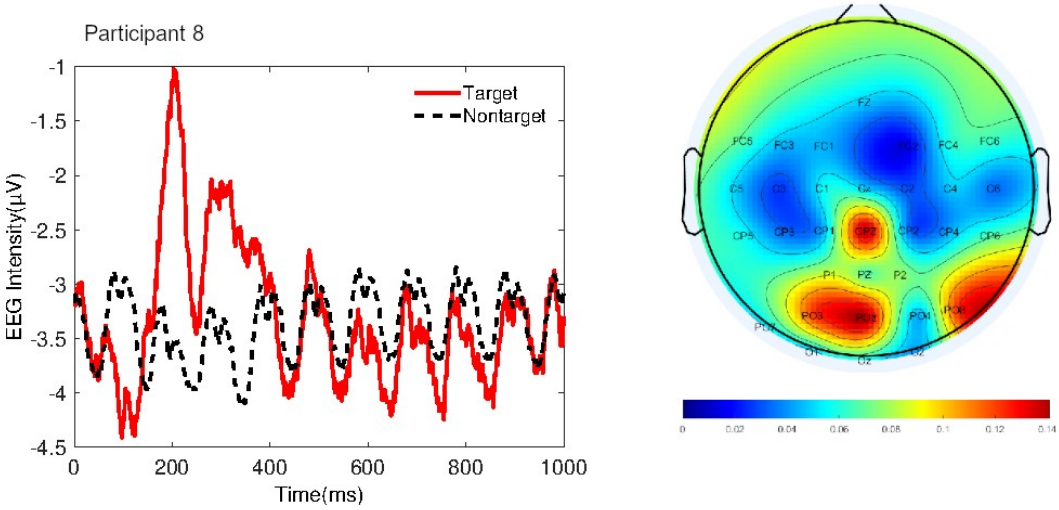

Participant 3
Target
Nontarget
EEG Intensity(μV)
Time(ms)
6
5
4
3
2
1
0
-1
0
200
400
600
800
1000

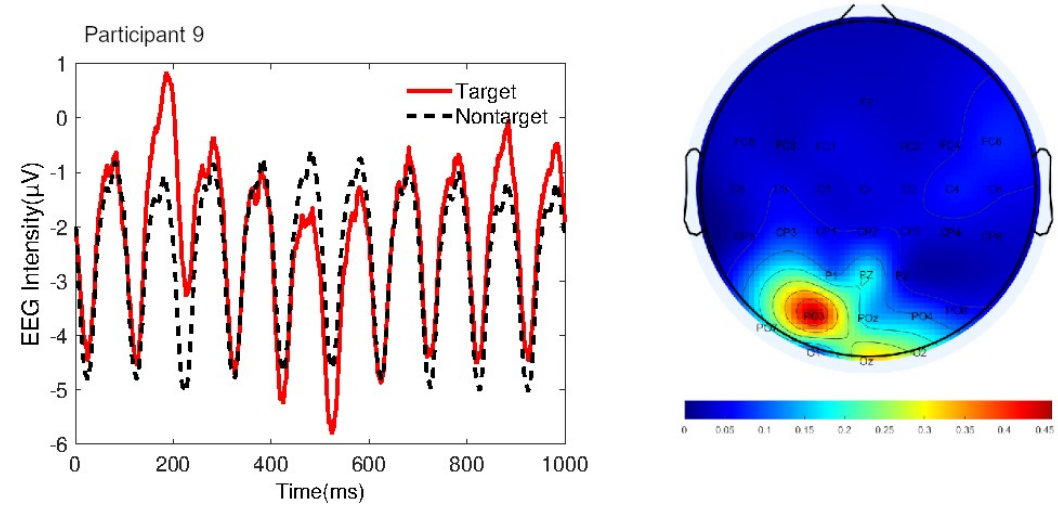

Participant 4
Target
Nontarget
EEG Intensity(μV)
Time(ms)

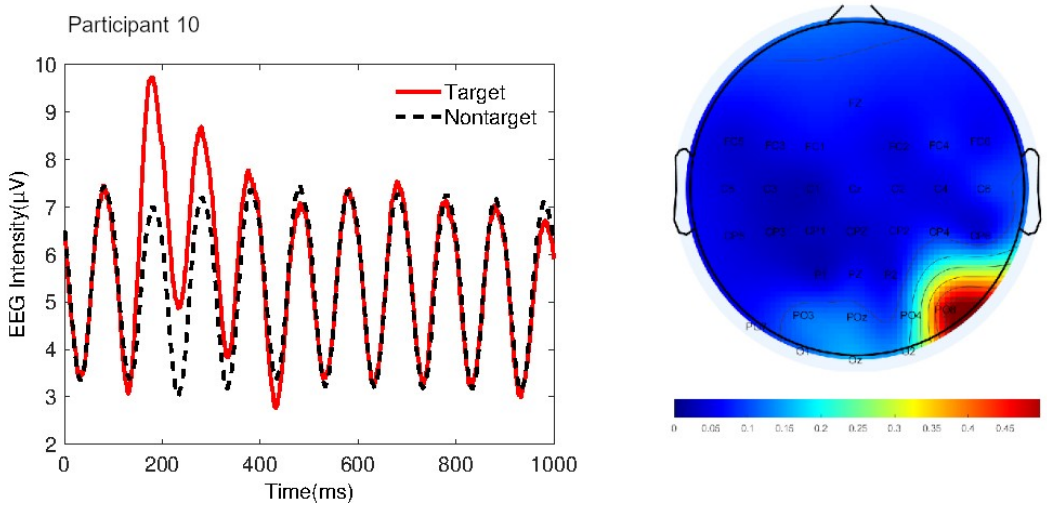

Participant 5
Target
Nontarget
EEG Intensity(μV)
Time(ms)

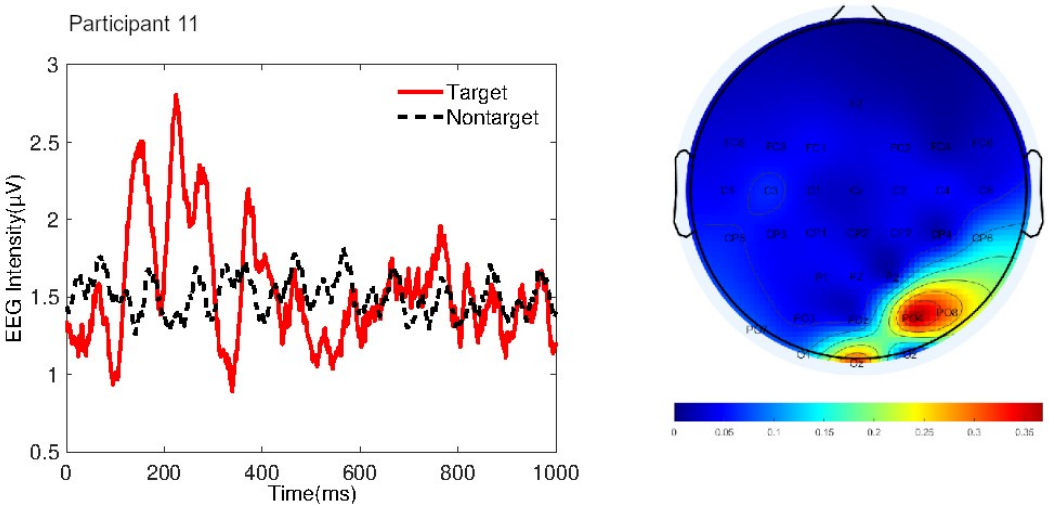

Participant 6
Target
Nontarget
EEG Intensity(μV)
Time(ms)

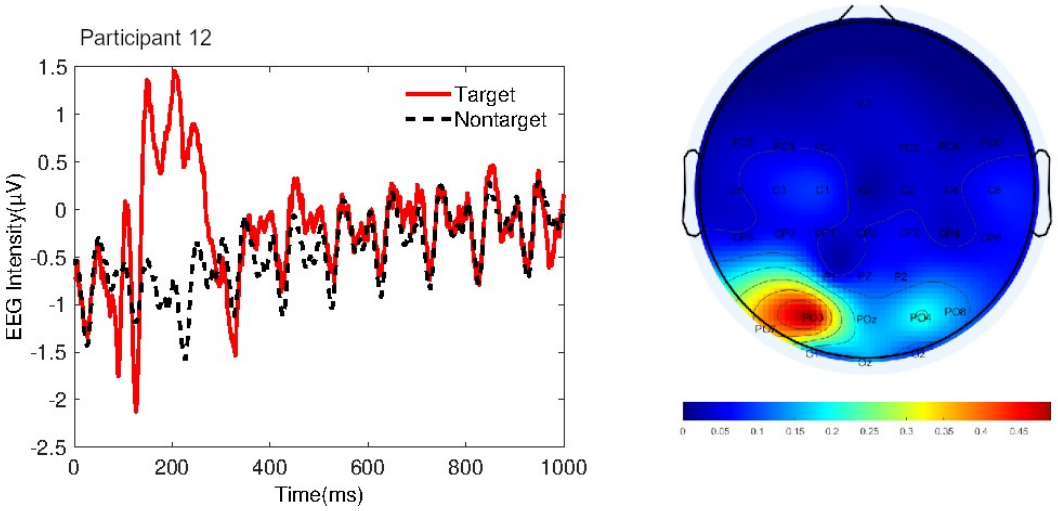

Participant 7
Target
Nontarget
EEG Intensity(μV)
Time(ms)

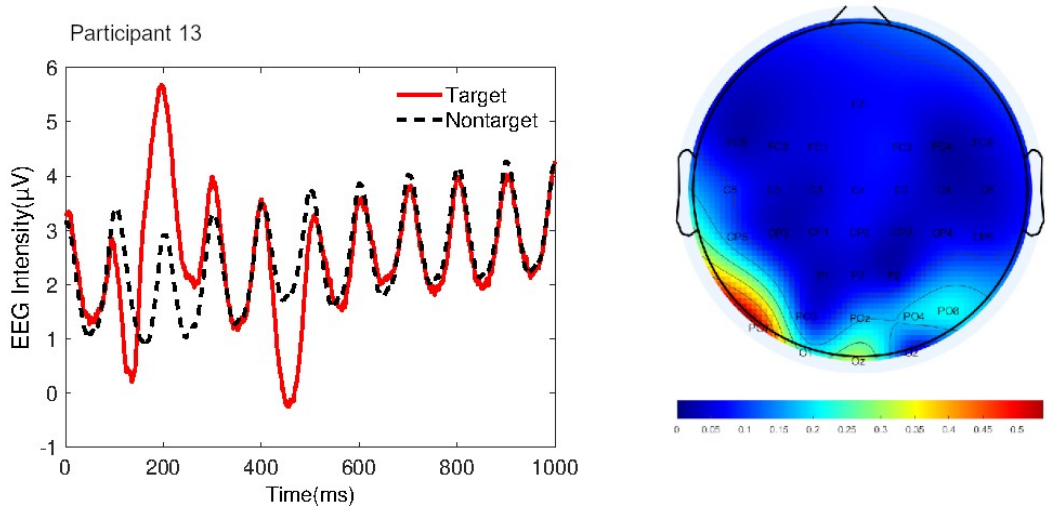

Participant 8
Target
Nontarget
EEG Intensity(μV)
Time(ms)

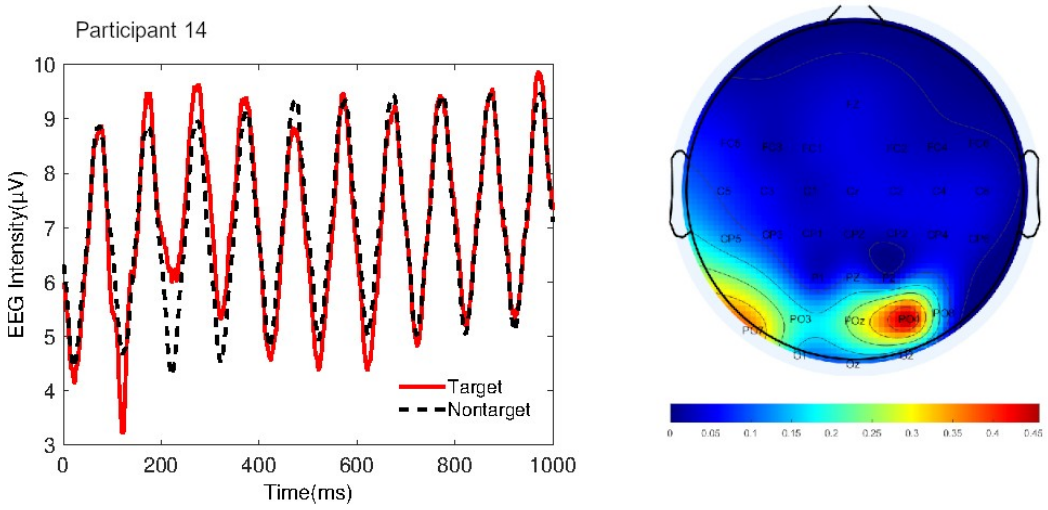

Participant 9
Target
Nontarget
EEG Intensity(μV)
Time(ms)

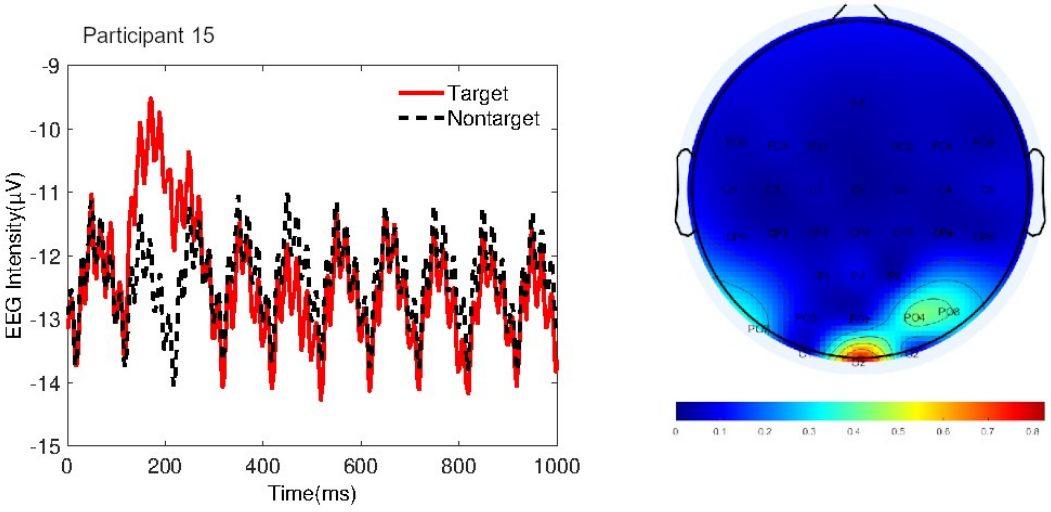

Participant 10
Target
Nontarget
EEG Intensity(μV)
Time(ms)


Participant 11
Target
Nontarget
EEG Intensity(μV)
Time(ms)


Participant 12
Target
Nontarget
EEG Intensity(μV)
Time(ms)


Participant 13
Target
Nontarget
EEG Intensity(μV)
Time(ms)


Participant 14
Target
Nontarget
EEG Intensity(μV)
Time(ms)


Participant 15
Target
Nontarget
EEG Intensity(μV)
Time(ms)

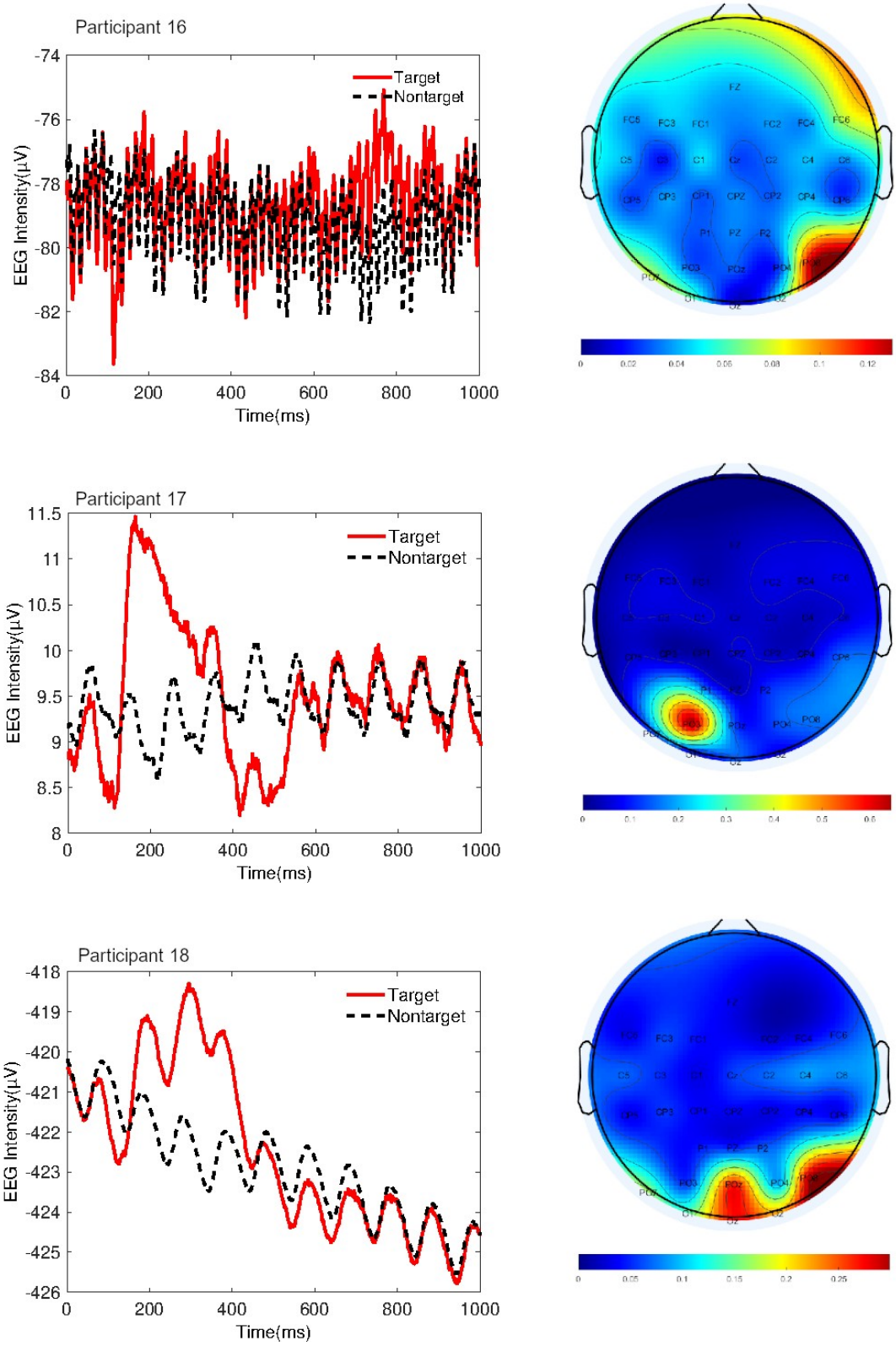



Figure 4.22 EEG plots and heatmaps

# 5. CONCLUSION

Video anomaly detection using RNN prediction methods are promising since RNN can use temporal connections of a video. We proposed multiple semi-supervised and unsupervised feature extraction methods. With these features we trained an RNN that can predict future features. Since anomalies are unexpected events RNN would fail to predict anomalous frames. However our study has several weaknesses. Improved video anomaly detection methods based on unsupervised RNN, can be done by addressing these weaknesses. C3D features can be replaced with other feature extraction methods. Summarizing each video in 32 parts eliminates the bias toward longer videos, however UCF-crime dataset contains videos ranging from 30 seconds to 8 hours. While shorter videos can be expressed with fewer parts, 32 parts is not enough to represent longer videos. A different feature extraction method that can express longer videos with more features without introducing bias, can be used. Furthermore to improve results, RNN and feature extraction methods can be trained together.

In this thesis, we introduced a new electroencephalogram (EEG) dataset generated by a P300-based brain computer interface (BCI) system utilizing the row-column (RC) paradigm. The introduced dataset has certain advantages compared to the existing publicly available P300-based BCI speller datasets. First, the scale of our dataset is relatively large as it includes 18 subjects, and 10 blocks of EEG data per subject. Second, it presents the character matrix in $5 \times 8$ form which is in line with the aspect ratio of most monitors in use, whereas other P300 datasets typically use the square $6 \times 6$ matrix. Also, the $5 \times 8$ character matrix presents 40 characters at once and hence has the potential to transfer more information. Third, and most importantly, in our speller experiments, we used significantly shorter stimulus duration (SD) and inter-stimulus intervals (ISI) to better represent the requirements of realistic high speed spellers. However, the rapid character presentation (i.e. short SD and short ISI) also calls for strong signal processing and machine learning challenges since now the detection of P300 is intrinsically complicated due to the P300 interference. To conclude, the introduced dataset is challenging and expected to encourage

further P300 detection studies (from the machine learning and signal processing perspective) for addressing the realistic requirements of high-speed P300-based BCI spellers using the RC paradigm.

# BIBLIOGRAPHY


Brain Products GmbH - Solutions for neurophysiological research. `https://www.brainproducts.com/index.php`. Accessed: 2021-03-26.

C Guan , Thulasidas, M., & J Wu (2004). High performance P300 speller for brain-computer interface. In *IEEE International Workshop on Biomedical Circuits and Systems, 2004.*, (pp. S3/5/INV–S3/13).

Adam, A., Rivlin, E., Shimshoni, I., & Reinitz, D. (2008). Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(3), 555–560.

Ahi, S. T., Kambara, H., & Koike, Y. (2011). A Dictionary-Driven P300 Speller With a Modified Interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *19*(1), 6–14.

Alhaddad, M. J. (2012). Common average reference (CAR) improves P300 speller. *International Journal of Engineering and Technology*, *2*(3), 21.

Belitski, A., Farquhar, J., & Desain, P. (2011). P300 audio-visual speller. *Journal of neural engineering*, *8*, 025022.

Blankertz, B., Muller, K. R., Curio, G., Vaughan, T. M., Schalk, G., Wolpaw, J. R., Schlogl, A., Neuper, C., Pfurtscheller, G., Hinterberger, T., Schroder, M., & Birbaumer, N. (2004). The BCI competition 2003: progress and perspectives in detection and discrimination of EEG single trials. *IEEE Transactions on Biomedical Engineering*, *51*(6), 1044–1051.

Blankertz, B., Muller, K. R., Krusienski, D. J., Schalk, G., Wolpaw, J. R., Schlogl, A., Pfurtscheller, G., Millan, J. R., Schroder, M., & Birbaumer, N. (2006). The BCI competition III: validating alternative approaches to actual BCI problems. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *14*(2), 153–159.

Boiman, O. & Irani, M. (2005). Detecting irregularities in images and in video. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, (pp. 462–469 Vol. 1).

Brunner, P., Joshi, S., Briskin, S., Wolpaw, J. R., Bischof, H., & Schalk, G. (2010). Does the 'P300' speller depend on eye gaze? *Journal of Neural Engineering*, *7*(5), 056013.

Cecotti, H. & Graser, A. (2011). Convolutional Neural Networks for P300 Detection with Application to Brain-Computer Interfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(3), 433–445.

Cecotti, H. & Graser, A. (2011). Convolutional Neural Networks for P300 Detection with Application to Brain-Computer Interfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(3), 433–445.

Chandola, V., Banerjee, A., & Kumar, V. (2016). *Anomaly Detection*, (pp. 1–15). Boston, MA: Springer US.

Chaurasiya, R. K., Londhe, N. D., & Ghosh, S. (2016). A Novel Weighted Edit Distance-Based Spelling Correction Approach for Improving the Reliability of Devanagari Script-Based P300 Speller System. *IEEE Access*, *4*, 8184–8198.

Colwell, K. A., Ryan, D. B., Throckmorton, C. S., Sellers, E. W., & Collins, L. M. (2014). Channel selection methods for the p300 speller. *Journal of Neuro-*

*science Methods*, *232*, 6 – 15.

da Costa, K. A., Papa, J. P., Passos, L. A., Colombo, D., Ser, J. D., Muhammad, K., & de Albuquerque, V. H. C. (2020). A critical literature survey and prospects on tampering and anomaly detection in image data. *Applied Soft Computing*, *97*, 106727.

Daly, J. J. & Wolpaw, J. R. (2008). Brain–computer interfaces in neurological rehabilitation. *The Lancet Neurology*, *7*(11), 1032 – 1043.

Dietterich, T. G., Lathrop, R. H., & Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, *89*(1), 31–71.

Doshi, K. & Yilmaz, Y. (2020). Fast unsupervised anomaly detection in traffic videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Dubey, S., Boragule, A., & Jeon, M. (2020). 3d resnet with ranking loss function for abnormal activity detection in videos. *CoRR*, *abs/2002.01132*.

Duncan-Johnson, C. C. & Donchin, E. (1977). On quantifying surprise: The variation of event-related potentials with subjective probability. *Psychophysiology*, *14*(5), 456–467.

Duvinage, M., Castermans, T., Petieau, M., Seetharaman, K., Hoellinger, T., Cheron, G., & Dutoit, T. (2012). A subjective assessment of a P300 BCI system for lower-limb rehabilitation purposes. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, (pp. 3845–3849).

Eidel, M. & Kübler, A. (2020). Wheelchair Control in a Virtual Environment by Healthy Participants Using a P300-BCI Based on Tactile Stimulation: Training Effects and Usability. *Frontiers in Human Neuroscience*, *14*, 265.

Farwell, L. A. & Donchin, E. (1988). Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and Clinical Neurophysiology*, *70*(6), 510 – 523.

Fazel-Rezai, R. & Abhari, K. (2009). A region-based P300 speller for brain-computer interface. *Canadian Journal of Electrical and Computer Engineering*, *34*(3), 81–85.

F.R.S., K. P. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *2*(11), 559–572.

Guy, V., Soriani, M.-H., Bruno, M., Papadopoulo, T., Desnuelle, C., & Clerc, M. (2018). Brain computer interface with the P300 speller: Usability for disabled people with amyotrophic lateral sclerosis. *Annals of Physical and Rehabilitation Medicine*, *61*(1), 5 – 11.

Hara, K., Kataoka, H., & Satoh, Y. (2017). Learning spatio-temporal features with 3d residual networks for action recognition. *CoRR*, *abs/1708.07632*.

Hoffmann, U., Vesin, J.-M., Ebrahimi, T., & Diserens, K. (2008). An efficient p300-based brain–computer interface for disabled subjects. *Journal of Neuroscience Methods*, *167*(1), 115–125. Brain-Computer Interfaces (BCIs).

Ji, J., Porjesz, B., Begleiter, H., & Chorlian, D. (1999). P300: the similarities and differences in the scalp distribution of visual and auditory modality. *Brain topography*, *11*(4), 315–327.

Kabbara, A., Hassan, M., Khalil, M., Eid, H., & El Falou, W. (2015). An efficient

P300-speller for Arabic letters.

Kachenoura, A., Albera, L., Senhadji, L., & Comon, P. (2008). Ica: a potential tool for bci systems. *IEEE Signal Processing Magazine*, *25*(1), 57–68.

Kaplan, A. Y., Shishkin, S. L., Ganin, I. P., Basyul, I. A., & Zhigalov, A. Y. (2013). Adapting the P300-Based Brain–Computer Interface for Gaming: A Review. *IEEE Transactions on Computational Intelligence and AI in Games*, *5*(2), 141–149.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *CVPR*.

Kim, C.-M., Hong, E. J., & Park, R. C. (2021). Chest x-ray outlier detection model using dimension reduction and edge detection. *IEEE Access*, *9*, 86096–86106.

Kindermans, P. J., Verstraeten, D., & Schrauwen, B. (2012). A Bayesian Model for Exploiting Application Constraints to Enable Unsupervised Training of a P300-based BCI. *PLOS ONE*, *7*(4), 1–12.

Kingma, D. & Ba, J. (2014). Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Klem, G., Lüders, H., Jasper, H., & Elger, C. (1999). The ten-twenty electrode system of the international federation. the international federation of clinical neurophysiology. *Electroencephalography and clinical neurophysiology. Supplement*, *52*, 3–6.

Klimesch, W. (1999). EEG alpha and theta oscillations reflect cognitive and memory performance: a review and analysis. *Brain Research Reviews*, *29*(2), 169 – 195.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM*, *60*(6), 84–90.

Krusienski, D. J., Sellers, E. W., Cabestaing, F., Bayoudh, S., McFarland, D. J., Vaughan, T. M., & Wolpaw, J. R. (2006). A comparison of classification techniques for the P300 Speller. *Journal of Neural Engineering*, *3*(4), 299–305.

Krusienski, D. J., Sellers, E. W., McFarland, D. J., Vaughan, T. M., & Wolpaw, J. R. (2008). Toward enhanced P300 speller performance. *Journal of Neuroscience Methods*, *167*(1), 15 – 21. Brain-Computer Interfaces (BCIs).

Krusienski, D. J., Sellers, E. W., & Vaughan, T. M. (2007). Common Spatio-Temporal Patterns for the P300 Speller. In *2007 3rd International IEEE/EMBS Conference on Neural Engineering*, (pp. 421–424).

Kuo, C.-T. & Davidson, I. (2016). A framework for outlier description using constraint programming. *Proceedings of the AAAI Conference on Artificial Intelligence*, *30*(1).

Lee, J., Won, K., Kwon, M., Jun, S. C., & Ahn, M. (2020). CNN With Large Data Achieves True Zero-Training in Online P300 Brain-Computer Interface. *IEEE Access*, *8*, 74385–74400.

Lee, M.-H., Kwon, O.-Y., Kim, Y.-J., Kim, H.-K., Lee, Y.-E., Williamson, J., Fazli, S., & Lee, S.-W. (2019). EEG dataset and OpenBMI toolbox for three BCI paradigms: an investigation into BCI illiteracy. *GigaScience*, *8*(5). giz002.

Lee, S., Kim, H. G., & Ro, Y. M. (2018). STAN: spatio-temporal adversarial networks for abnormal event detection. *CoRR*, *abs/1804.08381*.

Li, H. & Boulanger, P. (2020). A survey of heart anomaly detection using ambulatory electrocardiogram (ecg). *Sensors*, *20*(5).

Liu, M., Wu, W., Gu, Z., Yu, Z., Qi, F., & Li, Y. (2018). Deep learning based on Batch Normalization for P300 signal detection. *Neurocomputing*, *275*, 288–297.

Lu, C., Shi, J., & Jia, J. (2013). Abnormal event detection at 150 fps in matlab. In *2013 IEEE International Conference on Computer Vision*, (pp. 2720–2727).

Lu, J., Speier, W., Hu, X., & Pouratian, N. (2012). The Effects of Stimulus Timing Features on P300 Speller Performance. *Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology*, *124*.

Luo, W., Liu, W., & Gao, S. (2017). A revisit of sparse coding based anomaly detection in stacked rnn framework. In *2017 IEEE International Conference on Computer Vision (ICCV)*, (pp. 341–349).

Mahadevan, V., Li, W., Bhalodia, V., & Vasconcelos, N. (2010). Anomaly detection in crowded scenes. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (pp. 1975–1981).

Mahadevan, V., LI, W.-X., Bhalodia, V., & Vasconcelos, N. (2010). Anomaly detection in crowded scenes. (pp. 1975–1981).

Mason, S. G., Bashashati, A., Fatourechi, M., Navarro, K. F., & Birch, G. E. (2007). A Comprehensive Survey of Brain Interface Technology Designs. *Annals of biomedical engineering*, *35*(2), 137–169.

Minett, J., Zheng, H.-Y., Fong, M., Zhou, L., Peng, G., & Wang, W. (2012). A Chinese Text Input Brain–Computer Interface Based on the P300 Speller. *International Journal of Human-computer Interaction - IJHCI*, *28*, 472–483.

Mirghasemi, H., Fazel-Rezai, R., & Shamsollahi, M. B. (2006). Analysis of P300 Classifiers in Brain Computer Interface Speller. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, (pp. 6205–6208).

Mouraux, A. & Iannetti, G. D. (2008). Across-trial averaging of event-related eeg responses and beyond. *Magnetic resonance imaging*, *26*(7), 1041–1054.

Nijboer, F., Sellers, E. W., Mellinger, J., Jordan, M. A., Matuz, T., Furdea, A., Halder, S., Mochty, U., Krusienski, D. J., Vaughan, T., Wolpaw, J. R., Birbaumer, N., & Kübler, A. (2008). A P300-based brain–computer interface for people with amyotrophic lateral sclerosis. *Clinical Neurophysiology*, *119*(8), 1909 – 1916.

Nuwer, M., Comi, G., Emerson, R., Fuglsang-Frederiksen, A., Guérit, J., Hinrichs, H., Ikeda, A., Luccas, F., & Rappelsberger, P. (1999). Ifcn standards for digital recording of clinical eeg. the international federation of clinical neurophysiology. *Electroencephalography and clinical neurophysiology. Supplement*, *52*, 11–4.

Oralhan, Z. (2019). A New Paradigm for Region-Based P300 Speller in Brain Computer Interface. *IEEE Access*, *7*, 106618–106627.

Ortner, R., Prueckl, R., Putz, V., Scharinger, J., Bruckner, M., Schnürer, A., & Guger, C. (2011). Accuracy of a P300 Speller for Different Conditions: A Comparison.

Philip, J. T. & George, S. T. (2020). Visual P300 Mind-Speller Brain-Computer Interfaces: A Walk Through the Recent Developments With Special Focus on Classification Algorithms. *Clinical EEG and Neuroscience*, *51*(1), 19–33. PMID: 30997842.

Pires, G., Castelo-Branco, M., & Nunes, U. (2008). Visual P300-based BCI to

steer a wheelchair: A Bayesian approach. In *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, (pp. 658–661).

Pires, G., Nunes, U., & Castelo-Branco, M. (2012a). Comparison of a row-column speller vs. a novel lateral single-character speller: Assessment of BCI for severe motor disabled patients. *Clinical Neurophysiology*, *123*(6), 1168 – 1181.

Pires, G., Nunes, U., & Castelo-Branco, M. (2012b). Comparison of a row-column speller vs. a novel lateral single-character speller: assessment of BCI for severe motor disabled patients. *Clinical Neurophysiology*, *123*(6), 1168–1181.

Polich, J. (2007). Updating P300: An integrative theory of P3a and P3b. *Clinical Neurophysiology*, *118*(10), 2128 – 2148.

Pourhabibi, T., Ong, K.-L., Kam, B. H., & Boo, Y. L. (2020). Fraud detection: A systematic literature review of graph-based anomaly detection approaches. *Decision Support Systems*, *133*, 113303.

Rakotomamonjy, A., Guigue, V., Mallet, G., & Alvarado, V. (2005). Ensemble of SVMs for Improving Brain Computer Interface P300 Speller Performances. In Duch, W., Kacprzyk, J., Oja, E., & Zadrożny, S. (Eds.), *"Artificial Neural Networks: Biological Inspirations – ICANN 2005"*, (pp. 45–50)., Berlin, Heidelberg. Springer Berlin Heidelberg.

Ramachandra, B. & Jones, M. J. (2019). Street scene: A new dataset and evaluation protocol for video anomaly detection. *CoRR*, *abs/1902.05872*.

Rebsamen, B., Burdet, E., Guan, C., Zhang, H., Teo, C. L., Zeng, Q., Ang, M., & Laugier, C. (2006). A Brain-Controlled Wheelchair Based on P300 and Path Guidance. In *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006.*, (pp. 1101–1106).

Rohani, D. A., Sorensen, H. B. D., & Puthusserypady, S. (2014). Brain-computer interface using P300 and virtual reality: A gaming approach for treating ADHD. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, (pp. 3606–3609).

Salvaris, M. & Sepulveda, F. (2009). Visual modifications on the P300 speller BCI paradigm. *Journal of Neural Engineering*, *6*(4), 046011.

Santhosh, K. K., Dogra, D. P., & Roy, P. P. (2020). Anomaly detection in road traffic using visual surveillance: A survey. *ACM Comput. Surv.*, *53*(6).

Schapire, R. E. & Freund, Y. (2012). *Foundations of Machine Learning*, (pp. 23–52).

Sellers, E. W., Kubler, A., & Donchin, E. (2006). Brain-computer interface research at the university of south Florida cognitive psychophysiology laboratory: the P300 speller. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *14*(2), 221–224.

Shan, H., Liu, Y., & Stefanov, T. (2019). Ensemble of Convolutional Neural Networks for P300 Speller in Brain Computer Interface. In Tetko, I. V., Kůrková, V., Karpov, P., & Theis, F. (Eds.), *"Artificial Neural Networks and Machine Learning – ICANN 2019: Text and Time Series"*, (pp. 376–394)., Cham. Springer International Publishing.

Singh, D. & Mohan, C. K. (2019). Deep spatio-temporal representation for detection of road accidents using stacked autoencoder. *IEEE Transactions on Intelligent Transportation Systems*, *20*(3), 879–887.

Sodemann, A. A., Ross, M. P., & Borghetti, B. J. (2012). A review of anomaly

detection in automated surveillance. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(6), 1257–1272.

Speier, W., Arnold, C., & Pouratian, N. (2013). Evaluating True BCI Communication Rate through Mutual Information and Language Models. *PLOS ONE*, *8*(10), 1–9.

Spüler, M. (2017). A high-speed brain-computer interface (BCI) using dry EEG electrodes. *PLOS ONE*, *12*(2), 1–12.

Squires, N. K., Squires, K. C., & Hillyard, S. A. (1975). Two varieties of long-latency positive waves evoked by unpredictable auditory stimuli in man. *Electroencephalography and Clinical Neurophysiology*, *38*(4), 387 – 401.

Stojanovic, L., Dinic, M., Stojanovic, N., & Stojadinovic, A. (2016). Big-data-driven anomaly detection in industry (4.0): An approach and a case study. In *2016 IEEE International Conference on Big Data (Big Data)*, (pp. 1647–1652).

Sultani, W., Chen, C., & Shah, M. (2018). Real-world anomaly detection in surveillance videos. *CoRR*, *abs/1801.04264*.

Ten, C.-W., Hong, J., & Liu, C.-C. (2011). Anomaly detection for cybersecurity of the substations. *IEEE Transactions on Smart Grid*, *2*(4), 865–873.

Townsend, G., LaPallo, B. K., Boulay, C. B., Krusienski, D. J., Frye, G. E., Hauser, C. K., Schwartz, N. E., Vaughan, T. M., Wolpaw, J. R., & Sellers, E. W. (2010). A novel P300-based brain–computer interface stimulus presentation paradigm: Moving beyond rows and columns. *Clinical Neurophysiology*, *121*(7), 1109 – 1120.

Tran, D., Bourdev, L. D., Fergus, R., Torresani, L., & Paluri, M. (2014). C3D: generic features for video analysis. *CoRR*, *abs/1412.0767*.

Tschuchnig, M. E. & Gadermayr, M. (2021). Anomaly detection in medical imaging - a mini review. *ArXiv*, *abs/2108.11986*.

Tsiouris, , Pezoulas, V. C., Zervakis, M., Konitsiotis, S., Koutsouris, D. D., & Fotiadis, D. I. (2018). A long short-term memory deep learning network for the prediction of epileptic seizures using eeg signals. *Computers in Biology and Medicine*, *99*, 24–37.

Tziakos, I., Cavallaro, A., & Xu, L.-Q. (2010). Local abnormality detection in video using subspace learning. In *2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, (pp. 519–525).

Vu, T.-H., Boonaert, J., Ambellouis, S., & Taleb-Ahmed, A. (2021). Multi-channel generative framework and supervised learning for anomaly detection in surveillance videos. *Sensors*, *21*(9).

Wang, J. & Xia, L. (2019). Abnormal behavior detection in videos using deep learning. *Cluster Computing*, *22*.

Wang, Y., Chen, X., Gao, X., & Gao, S. (2017). A benchmark dataset for ssvep-based brain–computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *25*(10), 1746–1752.

Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., & Vaughan, T. M. (2002). Brain–computer interfaces for communication and control. *Clinical Neurophysiology*, *113*(6), 767 – 791.

Xue, Y., Tang, J., He, F., Xu, M., & Qi, H. (2019). Improve P300 Speller Performance by Changing Stimulus Onset Asynchrony (SOA) Without Retraining the Subject-Independent Model. *IEEE Access*, *7*, 134137–134144.

Yuan, Y., Fang, J., & Wang, Q. (2015). Online anomaly detection in crowd scenes

via structure analysis. *IEEE Transactions on Cybernetics*, *45*(3), 548–561.

Zhong, J., Li, N., Kong, W., Liu, S., Li, T. H., & Li, G. (2019). Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection. *CoRR*, *abs/1903.07256*.