Automatic Detection of Mind-Wandering using Facial Expressions

Radek Kargul
Supervisors: Bernd Dudzik, Xucong Zhang, Hayley Hung
EEMCS, Delft University of Technology, The Netherlands
20-6-2022

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

Abstract

Spending time in front of screens has become an inescapable activity, which might be interrupted by unrelated external causes. While automatic approaches to identify mind-wandering (MW) have already been investigated, past research was done with self-reports or physiological data. This work explores automated detection utilizing solely facial expressions from Mementos data, which comes in the form of webcam recordings, where participants react to music videos. The recordings are annotated with labels indicating perceived MW. Video responses are turned into time series by first extracting facial characteristics, which are encoded with Facial Action Coding System (FACS). Temporal information is represented with 170 temporal features. Classification is conducted with support vector machines (SVM) through a data-level approach and an algorithm-level approach, first by synthesizing data and second by adding class weights to SVM. Both approaches are evaluated with metric scores insensitive to imbalanced data. On average, results show that detection performs marginally better than by chance. However, the evaluation metric values vary across multiple classification runs, thus the prospect of using the Mementos dataset for automatic MW detection based on only facial expressions is not promising.

1 Introduction

Paying attention to a performed action can be hard, as frequently an individual's mind drifts elsewhere and ceases concentrating on the present activity. Whether one is reading articles, watching videos, or following lectures online, the mind might refuse to concentrate on the present task for unanticipated periods. This phenomenon is known as mind-wandering (MW). There are numerous definitions of MW, but for the purpose of this study, it can be assumed that "when mind-wandering occurs, the executive components of attention tend to move away from the main activity" [1], not due to external factors or the person interacting with the external environment.

MW is seen within various daily activities. For the last two years, most universities have been offering online education due to the COVID-19 pandemic [2]. Instead of engaging in classroom activities, students attended lectures online [3]. MW on a frequent basis may lead to poor performance in fundamental activities such as online learning [4]. However, it may also occur when completing other activities (e.g., viewing movies, digital reading) [5]. As a consequence, the capacity to identify it in videos automatically might possibly increase the main task performance [6].

Most computers and smartphones are equipped with cameras, making video a good input for the detection system. Firstly, this research examines perceived MW rather than self-reported, as just video input helps make judgments in detection, reducing a need for extra data. Choosing only facial expressions for MW detection is a reasonable option, due to many successful applications of facial filters in social media [7]. Secondly, locating a publicly available dataset with sensitive data is fairly tough. The project's supervisors recently completed a research, and as a consequence, Mementos dataset [8] was generated and employed for this study. Finally, another publicly accessible dataset was examined [9], however it contains data referenced from original sources, perhaps rendering certain samples no longer available.

The goal of this study is to establish whether automatic detection of mind-wandering using only facial expressions from the Mementos dataset [8] can perform better than by chance. The video recordings utilized for the study originate from the Mementos dataset, which is the first multimodal corpus for computational modeling of emotion and memory processing in response to video material [8]. The data, acquired through webcams, does not include labels for MW and requires prior manual labeling. This leads to the following sub-questions, which the study aims to address:

1. How does dataset choice affect MW detection?
2. How does manual labeling of perceived MW affect its detection?
3. How well can perceived MW detection differentiate between MW and not-MW instances?

The research paper is divided into the following sections: Existing MW detection approaches are discussed in section 2. Section 3 describes the methodology and experiment setup. Results are presented and analyzed in section 4, while discussion and limitations can be found in section 5. Responsible research is discussed in section 6 and section 7 contains conclusions and possible extensions to this research.

2 Related Work

There is existing research on the detection of MW using facial features. The vast majority of works combine facial features with other types of data, such as motion tracking [10]. However, there is more interest in detecting mind-wandering through eye gaze [6] and physiology [11, 12]. Most studies also utilize self-reports collected from participants, which report MW rather than perceived MW [10–14]. The majority of research is not directly relevant to this study. While there is some interest in detecting it, most approaches consider more than just visual data.

A common method of MW detection is by using physiological measures. One approach was investigated, where participants were not assessed subjectively with self-reports or thought-probes only, to indicate MW periods they experienced [11]. The interest lies in combining the methods with the simultaneous control of respiration and fingertip pressure. Participants are asked to control those variables, where both are measured to check for synchronization that serves as an objective index for MW detection. Another research was performed in a more controlled environment with the help of biosignals [12]. Participants were engaged in mindfulness-based training, lasting five days. Various biosignals were measured to effectively determine MW events with 85% accuracy. As an outcome of the research, a mobile application was created for automatic MW detection. Other groups used electrophysiological signals and self-reports to build ML models for predicting MW state detection [13], resulting in an above-chance MW detection.

A more relevant study with facial features was performed [10]. However, self-report techniques were utilized to help in measuring MW. Face videos were recorded to extract different granularity levels, among which facial action units (AUs) were used. SVM models achieved 25.4% and 20.9% above-chance scores. Moreover, another facial feature-based study with self-reports was performed [14]. It utilized computer vision to extract facial features and body movements from videos. By using supervised machine learning, it achieved detection that is 31% over a chance model.

Additionally, most approaches, if not all, use self-reports as the ground truth for detecting mind-wandering, whereby the person in the test indicates when they catch themselves mind-wandering [15]. Furthermore, as useful and beneficial as using additional data alongside webcam videos for mind-wandering detection might be [11–13], it comes with limitations. Physiological data has to be obtained in an appropriate environment and with proper sensors, which on a larger scale would be cost-intensive [16]. Therefore, if using only visual data helps in MW detection, then such detection has the potential to be more accessible and inexpensive.

3 Methodology

In this study, the automatic detection of mind-wandering is first addressed with data labeling and preprocessing. This step helps in establishing the ground truth and discarding futile video responses. Once it is prepared, facial expression features as well as temporal (time series) features are extracted. Prepared feature vectors then help train two machine learning models that aim to distinguish between MW and not-MW instances.

3.1 Mementos Dataset Labeling

The Mementos dataset consists of 1995 curated webcam recordings from 297 distinct individuals reacting to different chunks of music videos [8]. Each recording is 1 minute (± 10 seconds) long. The dataset has been curated, meaning the video resolution has been adjusted to 640 by 480 pixels, video responses outside of 50-70 second range have been removed, etc. [8]).

The goal of data annotation is to catch instances in video responses that indicate perceived mind-wandering. First, a subset of 549 video (out of 1995) recordings is selected for annotation, of which 495 are eligible for labeling. Due to time constraints, labeling all videos was not an option. Participants' primary task was to watch videos shown to them. The eligibility is determined by checking if a video response sabotages the ability to detect MW. This includes a participant walking out of webcam view range, or not watching the video they were assigned to watch, essentially reacting to external tasks.

Each video response contains time intervals labeled as either MW or not-MW. The Mementos dataset was not originally labeled for MW, but participants' self-reports exist showing when it happens. For the purpose of this study, self-reported data is ignored, as those are not a part of facial expressions. Together with all project members, a *rule book* (Table 1) is established to aid with labeling. Each rule describes what visual or audible indicators to look out for while annotating the webcam recordings. Given a video response contains at least one indicator described in the *rule book*, that video's interval is annotated. Figure 1 (partially blurred to prevent leaking sensitive data) shows an instance of perceived MW, as indicated by a "smile", which is indicated in Table 1. This suggests that each video might contain more than one interval of MW.

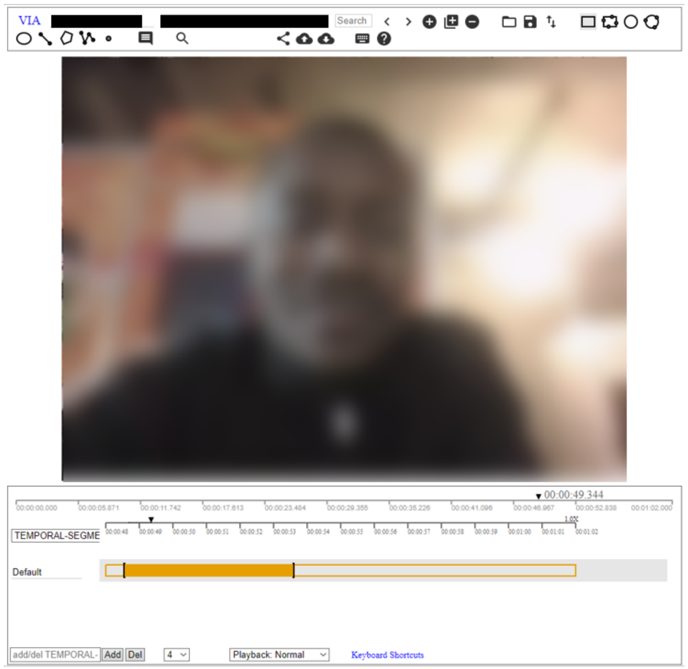


Figure 1: The VGG Annotator is used to annotate video recordings to find potential instances of perceived MW. The orange bar in the lower part of the figure indicates the start and end of MW in the video, while unmarked regions are not-MW. The selected region starts around the 48th second and ends after the 53rd second in the video.

VGG Image Annotator (VIA) [1] is an open source manual annotation software that was used to add annotations in this study. Figure 1 displays how annotation is done. To stay consistent with annotations, the group of five split into teams of two and three to annotate subsets of videos. This was performed to eliminate the influence of personal bias and to keep annotating uniform across all video samples. In cases of conflicts in decision-making, all five members participated in this process. A more robust approach was considered - *Fleiss' kappa*[2]. It measures agreements between raters to check whether there is agreement and whether decisions are objective. However, due to a limited time dedicated to dataset

[1] https://www.robots.ox.ac.uk/vgg/software/via/

[2] https://statistics.laerd.com/spss-tutorials/fleiss-kappa-in-spss-statistics.php

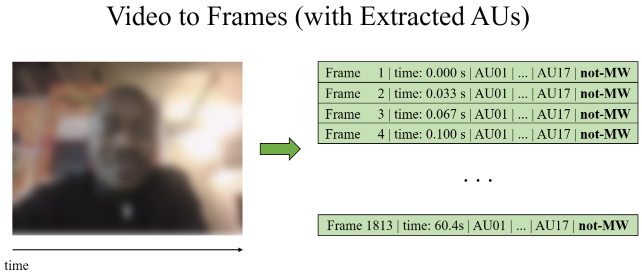
Sign	Description
Smile	Good memories - very expressive and sudden/genuine smile (reaction or response to the music video); subtle smile (a form of reminiscing/remembering a memory).
Looking up/Rolling eyes	Looking up should not be momentary and unrelated to primary task (not a distraction); eyes move from up to the side (remembering/recollecting)
Squinting eyes	Tendency to remember something/having some thoughts.
Person sounds	Person speaking/making a sound, unrelated to the song lyrics, but also not caused by an external stimuli.
Frown	Potential indication of bad/sad memories; a subtle frown (potential form of reminiscing/remembering a memory) but not very expressive and sudden (could be a reaction, or a response to the video).

Table 1: A rule book defined for annotating the dataset. Signs indicate perceived MW periods.

annotation, this approach was abandoned and group labeling persuaded instead.

3.2 Facial Expressions Feature Extraction

For MW detection, facial expressions are a term that could have various interpretations. This study defines them by the Facial Action Coding System (FACS) [17]. The expressions in this investigation serve as features, and their definition for this context is crucial. FACS was designed to target a multitude of media in detecting any muscle movements corresponding to facial expressions, which can be described with action units (AUs). These action units directly correspond to different singular movements of facial muscles. FACS hence helps in encoding facial expressions with descriptive precision [18]. The system has been used in many works that study facial expressions, where the AUs were used as features for classification [19, 20].



Frame 1 | time: 0.000 s | AU01 | ... | AU17 | **not-MW**
Frame 2 | time: 0.033 s | AU01 | ... | AU17 | **not-MW**
Frame 3 | time: 0.067 s | AU01 | ... | AU17 | **not-MW**
Frame 4 | time: 0.100 s | AU01 | ... | AU17 | **not-MW**
. . .
Frame 1813 | time: 60.4s | AU01 | ... | AU17 | **not-MW**

Figure 2: A video response split into frames through OpenFace 2.0. Each frame contains: frame number, frame timestamp, features (AU01-AU17) extracted, label (MW or not-MW). *Note: This figure serves as a toy example. It follows the data format used in this research.*

To extract AUs from webcam recordings, the OpenFace 2.0 [21] tool is used. It poses to be a common and reliable tool used in other works for facial feature extraction [8, 22, 23]. The tool detects 17 different action units, as listed in Table 2, and each has a description of the muscle it is associated with. The tool calculates these based on the intensity scale ranging from 1 (neutral state, muscle is not activated) to 5 (highly moved muscle). As Mementos dataset videos have 30 frames per second, OpenFace 2.0 generates extracts AUs for each frame (Figure 2). Therefore, the number of samples can range between 1500 and 2100, for a 50-second video and a 70-second video, respectively.

Code	Muscle Description
AU01	INNER BROW RAISER
AU02	OUTER BROW RAISER
AU04	BROW LOWERER
AU05	UPPER LID RAISER
AU06	CHEEK RAISER
AU07	LID TIGHTENER
AU09	NOSE WRINKLER
AU10	UPPER LIP RAISER
AU12	LIP CORNER PULLER
AU14	DIMPLER
AU15	LIP CORNER DEPRESSOR
AU17	CHIN RAISER
AU20	LIP STRETCHED
AU23	LIP TIGHTENER
AU25	LIPS PART
AU26	JAW DROP
AU45	BLINK

Table 2: List of AUs detected with OpenFace 2.0 in Mementos dataset.

The extracted action units carry information that needs to be carefully handled. There are multiple approaches that can be used to process the features. Intuitively, one can associate facial expressions with emotions (e.g. sadness, happiness, anger, fear, disgust, and surprise) [24] which are rather trivial to recognize with the human eye. However, it has been found that human emotions are not necessarily universal for humans [24], and for classification purposes, a more robust approach should be considered. Another possibility is to associate the 17 features with the rules defined for perceived mind-wandering. An issue with this approach is the loss of potentially useful information for the classification step. Assigning action units to the rules defined poses a subjective task. For instance, based on the units in Table 2, *smile* can correspond to a multitude of AU combinations. Hence, utilizing all action units extracted from the webcam recordings would prevent a loss of information and inherently help in classification.

3.3 Time Series Feature Extraction

Time series analysis helps to take temporal variations (or simply changes over time) in facial expressions into account in automatic detection. A time series is "a sequence of data points that occur in successive order over some period of time” [3]. It helps recognize patterns and trends in time in the form of features that can be extracted.

Classifying a singular frame offers no temporal information, therefore requiring a preprocessing step for time series feature extraction. Accounting for time series is tricky in this particular dataset, as the labeled MW instances are of variable lengths. Annotations of mind-wandering instances in the Mementos dataset vary between approx. 1 second (30 samples) and 10.4 seconds (312 samples) in length. Therefore, creating time series of equal lengths is not possible. Before clustering frames into consecutive segments, it is important to not create segments that contain frames with two distinct labels (i.e., the beginning of a MW instance is not clustered with previous frames that indicate not-MW). To ensure that such segmenting does not occur, video frames are first clustered into consecutive blocks of uniform label type. In Figure 3 such a concept is depicted, where (a) Video Response 1 does not have any MW instances, hence only 1 block of consecutive not-MW frames exists. On the other hand, (b) Video Response 2 contains one MW instance captured by frames 1324–1582. The video is hence split into 3 blocks, where blocks 1 and 3 contain consecutive not-MW frames each, and block 2 contains consecutive MW frames.

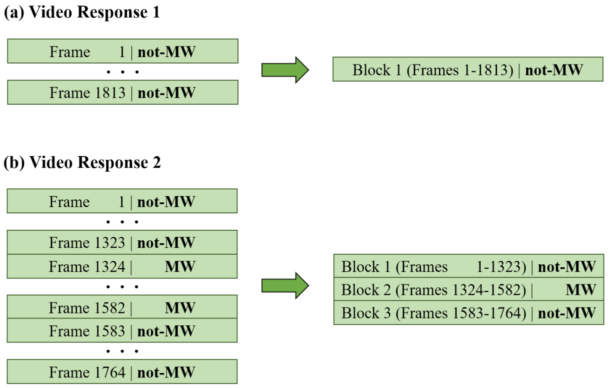


Figure 3: Frames to Blocks - (a) Video Response 1 consists of a sequence of 1813 Frames labeled as not-MW. That sequence is represented as Block 1 (b) Video Response 2 consists of 3 sequences of frames: Frames 1-1323 labeled as not-MW, Frames 1325-1582 as MW and Frames 1583-1764 as not-MW. Each sequence is represented as a block. *Note: This figure serves as a toy example. Some information in the figure is omitted.*

Each block holds a different number of frames. To retrieve temporal information from the video responses, blocks first need to be divided into shorter segments. To form them, each block, irrespective of its length (in frames), is divided by 156 frames. In Figure 4a, Block 1 (1813 frames) is divided into 12 segments (11 segments, each 156 frames and 1 segment with 97 frames). The segment length is established by taking the maximum MW instance in the dataset, and halving it. The chosen method ensures that each instance of MW is split into at most 2. This implies that there can also be segments with fewer than 156 frames (e.g. the 12th segment in Figure 4a). There might be better ways to create segments, but the main goal is to ensure segments contain consecutive frames with the same label and their lengths do not differ by more than 5.2 seconds (as opposed to 10.4 seconds if all MW instances were taken as a whole).

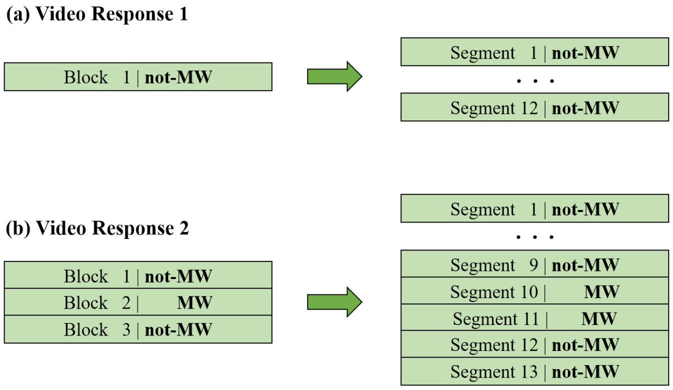


Figure 4: Blocks to Segments - (a) Video Response 1 contains only 1 sequence of frames with a uniform label of 0; 1 block is created holding that sequence. (b) Because Video Response 2 contains one MW instance (frames 1324–1582), three blocks are created for each consecutive sequence. *Note: This figure serves as a toy example. Some information in the figure is omitted.*

Time series feature extraction is done with *tsfresh*, a Python package allowing time series feature extraction which utilizes over 60 time series characterization methods and computes over 700 time series features [25]. It captures temporal information. *Tsfresh* is powerful in feature extraction for time series. For each time interval, it can extract up to 76 features that capture time series. The package offers three main feature extraction settings[4]: minimal, efficient, and comprehensive. The settings indicate what set of temporal features is computed for, in this study, action units.

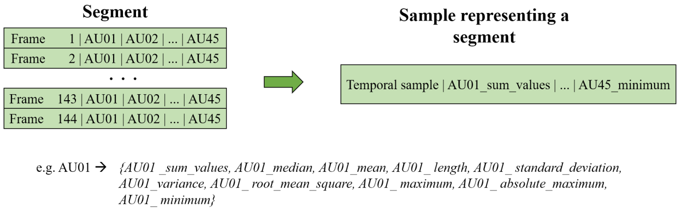

Segment
Frame 1 | AU01 | AU02 | ... | AU45
Frame 2 | AU01 | AU02 | ... | AU45
. . .
Frame 143 | AU01 | AU02 | ... | AU45
Frame 144 | AU01 | AU02 | ... | AU45
Sample representing a segment
Temporal sample | AU01_sum_values | ... | AU45_minimum
e.g. AU01 →
{AU01 _sum_values, AU01_median, AU01_mean, AU01_ length, AU01_ standard_deviation, AU01_variance, AU01_ root_mean_square, AU01_ maximum, AU01_ absolute_maximum, AU01_ minimum}

Figure 5: Temporal features extraction using *tsfresh* with minimal settings applied. A segment consisting of frames serves as an input for feature extraction. The output is a temporal sample with a new set of temporal features. *Note: This figure serves as a toy example. Some information in the figure is omitted.*

After investigating the settings, the minimal set of features is chosen. The set consists of 10 temporal features: *sum_values*, *median*, *mean*, *length*, *standard_deviation*, *variance*, *root_mean_square*, *maximum*, *absolute_maximum*, *minimum*. There are a number of reasons for its choice. First is feature space dimensionality. By providing 17 AUs, after

[3] https://www.investopedia.com/terms/t/timeseries.asp

[4] https://tsfresh.readthedocs.io/en/latest/text/list_of_features.html

temporal feature extraction, the final feature space sizes are 170, 13311, and 13413, for minimal, efficient, and comprehensive settings, respectively. Such large dimensionality generally increases classifier complexity, requiring reduction of the space before classification so as not to overfit the training set. Second, 10 features do not capture nearly as much information as 74 or 76, but 100 times larger feature spaces require more time to understand them, which is a large limitation for the duration of this study. Third, both efficient and comprehensive extraction methods generate NaN (Not a Number) values, which requires the removal of features to which those NaNs belong. This implication leads to poorly motivated feature space reduction in avoidance of NaNs. Lastly, large feature spaces make classification evaluation more ambiguous, potentially making drawn conclusions on classification more unjustifiable.

Figure 5 displays how temporal features are extracted from the prepared segments. Each segment is converted into a single sample. Each sample has a new set of 170 temporal features. All new samples are ready to be used for training a model and further classification on unseen data to evaluate the detection of mind-wandering.

3.4 Evaluation Metrics

After temporal feature extraction, there are 6011 temporal data samples, with 5945 and 66 labeled as not-MW and MW, respectively. This results in an approx. 90:1 ratio of not-MW to MW classes. This split is highly imbalanced and hence needs additional preparation. It is important to choose appropriate metrics when evaluating the performance of classifiers, as not all represent evaluation correctly when classifying imbalanced data.

	MW	not-MW
predicted MW	*TP* *(true positive)*	*FP* *(false positive)*
predicted not-MW	*FN* *(false negative)*	*TN* *(true negative)*
counts	MW_c	nMW_c

Table 3: Confusion matrix for two-class classification of MW and not-MW classes.

In two-class classification, a common metric for error estimation is accuracy. It measures the number of correctly classified samples. Given a confusion matrix (Table 3) (used for evaluation purposes), accuracy can be defined as

$$accuracy = \frac{TP + TN}{MW_c + nMW_c} \tag{1}$$

where the numerator accounts for correctly classified samples (MW and not-MW out of all samples) out of the total sample count. The complement of accuracy is error, which measures the number of incorrectly classified samples.

$$error = 1 - accuracy \tag{2}$$

However, both metrics lead to a dilemma since the not-MW class is significantly bigger (approx. 98.9%) compared to the MW class (approx. 1.1%). Here, the majority class impacts the score by a considerable number of properly categorized FP samples. Given that only a small fraction of the data identified MW, this metric fails to represent the reality that practically none of the samples from the minority class (MW) are recognized. In the confusion matrix, the left column represents the positive class, while the right column represents the negative class. According to [26], any metric that takes both values from the columns would be oblivious to the data imbalance. This explains the weak minority class representation, as (1) uses TP and FP, which do not belong to one column.

Other evaluation metrics are used to account for highly imbalanced data problems. As opposed to (1) and (2), recall evaluates the classification of true positives rate (describing how well the MW class was predicted) as well as false negatives rate (describing how well the not-MW class was predicted). The former one is referred to as sensitivity (3) and the latter as specificity (4).

$$sensitivity = recall = \frac{TP}{TP + FN} \tag{3}$$

$$specificity = \frac{TN}{FP + TN} \tag{4}$$

Similar to recall is precision (5), which summarizes the fraction of samples assigned to a MW class that actually belong to that MW class. Both metrics can be combined into an F-score, which represents a balance between (3) and (5).

$$precision = TP/(TP + FP) \tag{5}$$

Finally, the ROC Curve is a popular metric used for measuring how models are capable of distinguishing between classes. Nevertheless, this metric can be too optimistic when the minority class is small. To account for this, the Precision-Recall Curve is considered due to its focus on the minority class. However, comparing different models with the PR Curve is difficult, and hence the PR AUC (Area Under Curve) is used to have a numerical score. Therefore, all metrics that account for data imbalance are used for evaluation.

3.5 Empirical Investigation

Preparing a model for highly imbalanced time series data from the Mementos dataset is a difficult task, which requires numerous decisions to be made. Dealing with test/train split, choosing whether to over/under-sample the data, and what classifiers to choose are some of them. Because there are many possible ways to go about creating a model for this investigation, the following steps are implemented: Firstly, it is important to carefully prepare training and testing sets to ensure segments from the same participant do not occur in both training and testing sets. Secondly, choosing classifiers is also difficult because, given many choices, the study duration does not permit for thorough investigation. Lastly, data imbalances can be addressed with sampling techniques, which aim to restore the class proportions.

Data needs to be split into training and testing sets to train a classifier but prevent it from overfitting. There are many ways to perform data split, and the most common is k-fold cross-validation. It is a technique that splits a dataset into k-1 train sets, where 1 is reserved as a test set. When the data is highly imbalanced, this is not ideal because most train sets will only have the majority class samples, creating bias towards the not-MW label. A stratified version of the technique accounts for the imbalance. Stratified k-fold cross-validation ensures that each fold contains samples from both classes, making all folds more representative of both classes. In the Mementos dataset, there are only 1.1% MW samples, making stratified k-fold cross validation problematic. There are 78 participants, and each has between 1 and 7 response videos. Each response video is a set of prepared temporal segments. Not all responses and not all participants have at least 1 instance of mind-wandering. Samples belonging to one participant are quite similar, hence they need to be put in the same folds. There is a set of 33 participants, where each participant has at least 1 instance of MW, and a set of 45 that do not have any. From each set of random data, 70% of the data is used as training and 30% as testing. This ensures that each set has an equal number of class samples in both sets.

The training set is still imbalanced, but it can be modified without affecting the test data. Although there exist metrics that help evaluate models trained on imbalanced datasets, there also exist techniques that aim at reducing the imbalance in the training dataset. The first technique is random undersampling of the majority class, where samples from the negative class (not-MW) are removed to match the size of the minority class. This method removes most of the available data, not leaving enough samples for training. The second method is random oversampling of the minority class, where the minority class samples are replicated with replacement, increasing the size to match the majority class. This method creates identical duplicates, leading the training process to cause an overfit, increasing the computation time and decreasing classifier performance. An adapted oversampling method is used, SMOTE. It is a synthetic minority oversampling technique that creates new minority class samples based on the existing ones but does not duplicate them. At first, it selects an instance from the minority class at random and finds its k-nearest neighbors. The instance is connected to those k neighbors to form a segment and new synthetic samples are created in the feature space [27]. Although this method is not ideal, as it creates synthetic samples that are not real and hence not 100% representative of the minority class, it results in less overfitting and generally performs better than other sampling techniques [28].

To perform classification, the Python scikit-learn[5] library is used, which is a well-known and widely used tool for machine learning implementation. The data is prepared to be trained on a binary (two-class) classifier. A support vector machine (SVM) is selected, as it has an option to assign weights to classes and give more importance to the minority class [27]. This is especially useful when training on imbalanced data. Additionally, a SVM without class-weighting is used on SMOTE-sampled data. A baseline classifier[6], which classifies the majority class, is used as a baseline for comparing the results. Two approaches are used to evaluate MW detection.

- **Data-level approach**: The training set proportions are adjusted by SMOTE, where the minority class is upsampled to match the majority. An SVM is trained on this adjusted training dataset.
- **Algorithm-level approach**: A class-weighted SVM is used, where the training set proportions are used to assign weights. This ensures the minority class is not at a disadvantage when training the SVM.

Classification was run 10 and 100 times. However, both models are used 100 times to generate results, giving a more realistic outcome. To ensure their performance, SVMs are hyperparameter-tuned with the 3 important parameters using GridSearch[7]:

1. 'C': [0.1, 1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 500, 1000]
2. 'gamma': ['scale']
3. 'kernel': ['rbf']

Some parameter options are eliminated due to long training times. The 'rbf' kernel is selected, as it works well on small datasets. Sigmoid is preferred for neural networks, polynomial is less accurate than other kernels, and linear works well on high-dimensional data. A regularization parameter, C, indicates the margin between classes (the greater the value, the tighter the separation). Gamma is set to 'scale', as extensive search takes much more time. Lastly, for class-weighted SVM, weight on the minority class is put on according to the ratio of training samples to account for the difference.

4 Results

Evaluation is undertaken in line with the approach stated in section 3. Analysis 1 contrasts the performance of a class-weighted SVM (on an imbalanced dataset) versus an SVM (on synthetically balanced data) by checking how they compare to a baseline classifier. Analysis 2 offers a closer look into confusion matrices to understand the findings gained in Analysis 1.

4.1 Analysis 1: Classification Against the Baseliner

Analysis 1 is conducted with a classification performed on an oversampled training set and imbalanced training. All runs have a 70/30 train/test split, where proportions of MW and not-MW samples are preserved. Training and testing sets share no samples from the same participant.

Firstly, the baseline classifier achieves 0 precision and recall, as it only classifies the majority class, leaving the other unclassified, making it difficult to compare with other classifiers. On average, both techniques outperform the baseline classifier, where PR-AUC achieved by class-weighted is

[5] https://scikit-learn.org/stable/

[6] https://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyClassifier.html

[7] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

Baseline Classifier					
Metric	**Average**	**Median**	**Min**	**Max**	**SD**
Precision	0.0	0.0	0.0	0.0	0.0
Recall	0.0	0.0	0.0	0.0	0.0
F1-Score	0.0	0.0	0.0	0.0	0.0
PR-AUC	0.505	0.505	0.503	0.508	0.001

Table 4: Metric scores for baseline classifier metric scores (based on 100 classifications).

SVM Classifier (class-weighted)					
Metric	**Average**	**Median**	**Min**	**Max**	**SD**
Precision	0.516	0.535	0.132	0.75	0.147
Recall	0.646	0.667	0.111	0.9	0.15
F1-Score	0.55	0.571	0.19	0.762	0.127
PR-AUC	0.583	0.588	0.266	0.778	0.102

Table 5: Metric scores for class-weighted SVM classifier using imbalanced dataset (based on 100 runs).

0.583, SVM with SMOTE is 0.587, and the baseline is 0.505. However, the variance between the minimum and maximum scores indicates a huge influence the minority class has on the classification. This could be prevented by having more samples in the test set with a MW label. However, the dataset did not contain enough samples to account for this.

SVM Classifier (SMOTE technique)					
Metric	**Average**	**Median**	**Min**	**Max**	**SD**
Precision	0.546	0.561	0.146	1.0	0.16
Recall	0.624	0.638	0.053	0.9	0.169
F1-Score	0.556	0.587	0.1	0.821	0.135
PR-AUC	0.587	0.6	0.202	0.822	0.114

Table 6: Metric scores for SVM classifier using dataset with SMOTE technique applied (based on 100 runs).

Secondly, there is a large variance present in the metric scores across the classifications. Precision values shown in Table 5 range between 0.132 and 0.75. A low precision value indicates that only a small fraction of MW samples were assigned to their class correctly. This leads to an observation that at the ratio of 90:1 with roughly 20 samples in a test set, even 1 incorrectly classified sample accounts for 5% of the positive class. Recall also varies a lot, with values between 0.111 and 0.9, implying the huge influence the minority class has due to misclassifications. The F1-score follows the same pattern as it is derived from both precision and recall.

Finally, the results of both classification methods summarized in Table 6 and 5 are comparable. There is a slight increased performance from SVM on a balanced training set, however, only varying by +0.06 and +0.04 for average F1-Score and PR-AUC, respectively. Standard deviation suggests that the worse performing method achieved a marginally less sparse set of scores across 100 runs. Median values also indicate that the majority of classifications performed above average, yet still leaving outliers, by looking at minimum and maximum scores.

4.2 Analysis 2: Confusion Matrix

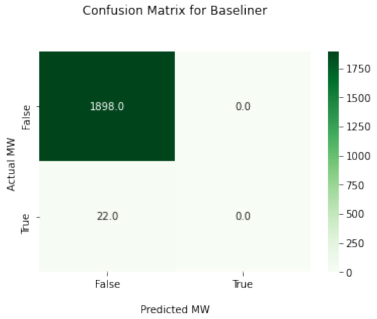


Figure 6: Confusion matrix Baseline Classifier with F1-Score=0.0

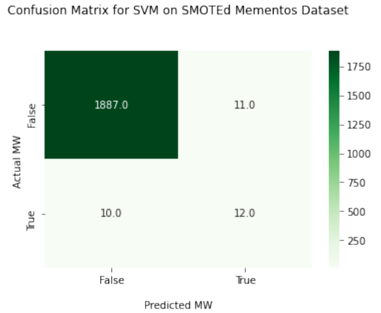


Figure 7: Confusion matrix for a SVM with SMOTE techqniue with F1-Score = 0.53

Confusion matrices assist in further evaluation and comprehension of the MW and not-MW samples' categorization. To understand the classification findings better, Figures 6, 8 and 7 display confusion matrices for one classification using the three classifiers from Analysis 1. This particular classification is a perfect example of showing how a single wrongly predicted positive sample affects the evaluation metrics. The evaluation test set has 1898 not-MW and 22 MW samples.

The baseliner correctly classified all samples as not-MW, resulting in a 0.0 F1-Score (Figure 6). SVM with SMOTE classified 12 MW samples correctly, giving a 54.5% success rate and achieving a 0.53 F1-Score (Figure 7). However, class-weighted SVM classified 1 MW sample more and that bumped its success rate to 59.0% and 0.57 F1-Score. This clearly explains how the class ratio affects classification. On the other hand, the majority class is successfully classified with a 99.4% success rate for both classifiers.

5 Discussion and Limitations

The results demonstrate that predicting perceived MW using the Mementos dataset is rather inconclusive. A few possible limitations are addressed that have an effect on the evaluation scores and perceived MW prediction.

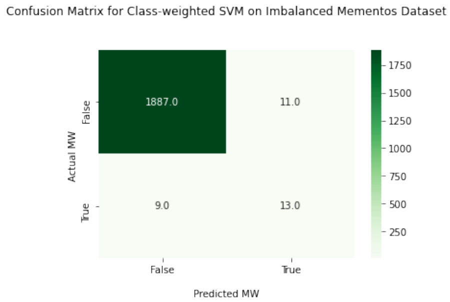


Figure 8: Confusion matrix for a class-weighted SVM with F1-Score = 0.57

The significant emphasis in the work is centered around class distribution for classification. Performing classification on a substantially skewed dataset is tricky in and of itself. With such a tiny number of samples, the identification of MW samples becomes a more connected challenge to the credit card fraud detection problem, where classification algorithms are relatively useless [29], especially in severely unbalanced datasets. This leads to a huge number of false positives, making mind-wandering detection imprecise. SVMs are thought to be less susceptible to class imbalance problems than other classifiers [30], but they can be ineffective in establishing the class boundary when the class distribution is skewed [31]. In addition, a dataset can also contain noisy data. In imbalanced datasets, this could imply difficulties in distinguishing between a rare case of a MW sample and a noisy not-MW sample [32].

The second inference arises from accounting for the dataset structure. The samples utilized in categorization originated from diverse participants, which required specific separation to avoid direct model overfitting from exposing data to the test set. Despite carefully separating data across training and testing sets, classification findings reveal that some did substantially better than others. It is quite plausible that applying other feature extraction methods would have resulted in different conclusions, which may be an intriguing addition to this research.

Another possible barrier arises from the annotation of perceived MW in the dataset. Videos were labeled using symptoms of reported mind-wandering according to Table 1. After a comprehensive analysis, there is no uniform evidence in other studies to substantiate the absolute veracity of the perceived MW indicators provided in this study. Perceived MW is not studied as extensively as MW detected with objective methods.

6 Responsible Research

The investigation of MW detection has been conducted in a responsible manner, with accordance to Netherlands Code of Conduct for Research Integrity (2018) [33]. The Mementos dataset contains highly sensitive data, namely webcam recordings of participants. To avoid exposing their sensitive information such as faces, figures used in this research are blurred. Additionally, the preprocessing of raw Mementos was handled offline. The annotation tool VGG allows local video labeling, without exposing data online. *Honesty* is addressed with explicit descriptions in methodology section which clearly outline used methods and approaches. All data preprocessing is explicitly described, where all steps are clearly described to avoid data fabrication. No misconduct nor datatrimming was practiced as all preprocessed data was utilized and no data was left out (from the data prepared for classification). Moreover, the results achieved suggest that *scrupulousness* and *transparency* are preserved, as results are rather inconclusive for using Mementos dataset for MW detection. According to [8] authors, the data itself was ethically collected and approved by the Human Research Ethics Committee of Delft University of Technology. The author of the paper takes full *responsibility* of presented content. The research is *independent*, as it does not use any tools that are commercially distributed or sponsor the author. Finally, the work is reproducible, as the methodology explicitly describes used tools and techniques, however the dataset cannot be accessed without Mementos authors' permission [8].

7 Conclusion and Future Work

The study aimed at resolving three sub-questions that arose: (1) including the choice of the dataset used to identify MW; (2) the influence of manually labeling the dataset; and (3) how well can perceived MW detection discriminate between MW and not-MW cases; as well as the main question, whether MW detection performs better than by chance.

Among the many limitations in the experiment, most indicate that the Mementos dataset is unsuitable for the detection. Firstly, the data balancing techniques addressed have no positive effect on the class imbalance. This leads to a question: what other approach can successfully balance a dataset? A possible direction is to use one-class classification, which turned out successful on exceptionally imbalanced data using SVM with highly dimensional feature space [34]. Other supervised work in one-class classification with autoencoders (deep learning) has also been successful [35]. However, deep learning requires a larger amount of data, which is not the case with the Mementos dataset. The Eev [9] dataset would be more suitable due to its larger size.

However, this leads to another implication of manually labeling data. First, labeling is substantially time-consuming, as it is equivalent to the total duration of considered videos. Although more footage could potentially result in a larger number of MW instances, this is practically impossible in this study due to time constraints. This implies that the number of MW instances collected is far too rare as opposed to the negative instances. Moreover, the inconclusive results indicated the impact that an imbalanced dataset can have on classification. Nevertheless, other methods of temporal feature retrieval could be considered. Utilizing a different set of temporal features for facial expression in combination with one-class classification could be considered, as it works well in high-dimensional spaces [34].

Nevertheless, among the reasons mentioned, perceived MW detection in this research is quite unsuccessful in differentiating the two labels. Another reason could be the extracted features. As only one set of temporal features was used, there is no comparison if other features could improve the classification. Other approaches propose tools to cope with time series data [36] by finding useful patterns in features.

Finally, the two classifications performed marginally better than the baseline implementation, but unfortunately, due to the data limitations, it is not possible to claim that MW detection is successful until evaluating the system on another dataset that contains more samples. Due to time constraints, the mentioned approaches could not be further explored in this research. However, they could lead to the construction of a more robust perceptual MW detection system that uses only facial expression.

References

[1] J. Smallwood and J. W. Schooler, "The restless mind." *Psychological bulletin*, vol. 132, no. 6, pp. 946–958, 2006.

[2] M. Maqableh and M. Alia, "Evaluation online learning of undergraduate students under lockdown amidst covid-19 pandemic: The online learning experience and students' satisfaction," *Children and Youth Services Review*, vol. 128, p. 106160, 2021.

[3] S. C. Pan, F. Sana, A. G. Schmitt, and E. L. Bjork, "Pretesting reduces mind wandering and enhances learning during online lectures," *Journal of Applied Research in Memory and Cognition*, vol. 9, no. 4, pp. 542–554, 2020.

[4] R. B. Hollis and C. A. Was, "Mind wandering, control failures, and social media distractions in online learning," *Learning and Instruction*, vol. 42, pp. 104–112, 2016.

[5] B. W. Mooneyham and J. W. Schooler, "The costs and benefits of mind-wandering: a review." *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale*, vol. 67, no. 1, p. 11, 2013.

[6] R. Bixler and S. D'Mello, "Automatic gaze-based user-independent detection of mind wandering during computerized reading," *User Modeling and User-Adapted Interaction*, vol. 26, no. 1, pp. 33–68, 2016.

[7] K. E. Anderson, "Getting acquainted with social networks and apps: Social media in 2017," *Library Hi Tech News*, 2017.

[8] B. Dudzik, H. Hung, M. A. Neerincx, and J. Broekens, "Collecting mementos: A multimodal dataset for context-sensitive modeling of affect and memory processing in responses to videos," *IEEE Transactions on Affective Computing*, 2021.

[9] J. J. Sun, T. Liu, A. S. Cowen, F. Schroff, H. Adam, and G. Prasad, "Eev: A large-scale dataset for studying evoked expressions from video," *arXiv preprint arXiv:2001.05488*, 2021.

[10] N. Bosch and S. K. D'mello, "Automatic detection of mind wandering from video in the lab and in the classroom," *IEEE Transactions on Affective Computing*, vol. 12, no. 4, pp. 974–988, 2019.

[11] Y. Zheng, D. Wang, Y. Zhang, and W. Xu, "Detecting mind wandering: an objective method via simultaneous control of respiration and fingertip pressure," *Frontiers in Psychology*, vol. 10, p. 216, 2019.

[12] M. Cheetham, C. Cepeda, H. Gamboa, J. Gilbert, H. Azhari, and A. Hesham, "Automated detection of mind wandering: A mobile application," 2016.

[13] H. W. Dong, C. Mills, R. T. Knight, and J. W. Kam, "Detection of mind wandering using eeg: Within and across individuals," *Plos one*, vol. 16, no. 5, p. e0251490, 2021.

[14] A. Stewart, N. Bosch, H. Chen, P. Donnelly, and S. D'Mello, "Face forward: Detecting mind wandering from video during narrative film comprehension," in *International Conference on Artificial Intelligence in Education*. Springer, 2017, pp. 359–370.

[15] A. Stewart, N. Bosch, and S. K. D'Mello, "Generalizability of face-based mind wandering detection across task contexts." *International Educational Data Mining Society*, 2017.

[16] R. Bixler, N. Blanchard, L. Garrison, and S. D'Mello, "Automatic detection of mind wandering during reading using gaze and physiology," in *Proceedings of the 2015 ACM on international Conference on Multimodal Interaction*, 2015, pp. 299–306.

[17] P. Ekman and W. V. Friesen, "Facial action coding system," *Environmental Psychology & Nonverbal Behavior*, 1978.

[18] Y.-I. Tian, T. Kanade, and J. F. Cohn, "Recognizing action units for facial expression analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 2, pp. 97–115, 2001.

[19] J. Grafsgaard, J. B. Wiggins, K. E. Boyer, E. N. Wiebe, and J. Lester, "Automatically recognizing facial expression: Predicting engagement and frustration," in *Educational data mining 2013*, 2013.

[20] J. Lien, T. Kanade, J. Cohn, and C.-C. Li, "Automated facial expression recognition based on facs action units," in *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, 1998, pp. 390–395.

[21] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L.-P. Morency, "Openface 2.0: Facial behavior analysis toolkit," in *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*. IEEE, 2018, pp. 59–66.

[22] X. Niu, H. Han, J. Zeng, X. Sun, S. Shan, Y. Huang, S. Yang, and X. Chen, "Automatic engagement prediction with gap feature," in *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, 2018, pp. 599–603.

[23] S. Chen, Z. Liu, J. Liu, Z. Yan, and L. Wang, "Talking head generation with audio and speech related facial action units," *arXiv preprint arXiv:2110.09951*, 2021.

[24] R. E. Jack, O. G. Garrod, H. Yu, R. Caldara, and P. G. Schyns, “Facial expressions of emotion are not culturally universal,” *.Proceedings of the National Academy of Sciences*, vol. 109, no. 19, pp. 7241–7244, 2012.

[25] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr, “Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package),” *Neurocomputing*, vol. 307, pp. 72–77, 2018.

[26] H. He and E. A. Garcia, “Learning from imbalanced data,” *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[27] R. Batuwita and V. Palade, “Class imbalance learning methods for support vector machines,” *Imbalanced learning: Foundations, algorithms, and applications*, pp. 47–50, 2013.

[28] G. Douzas, F. Bacao, and F. Last, “Improving imbalanced learning through a heuristic oversampling method based on k-means and smote,” *Information Sciences*, vol. 465, pp. 1–20, 2018.

[29] S. Makki, Z. Assaghir, Y. Taher, R. Haque, M.-S. Hacid, and H. Zeineddine, “An experimental study with imbalanced classification approaches for credit card fraud detection,” *IEEE Access*, vol. 7, pp. 93 010–93 022, 2019.

[30] N. Japkowicz and S. Stephen, “The class imbalance problem: A systematic study,” *Intelligent data analysis*, vol. 6, no. 5, pp. 429–449, 2002.

[31] G. Wu and E. Y. Chang, “Class-boundary alignment for imbalanced dataset learning,” in *ICML 2003 workshop on learning from imbalanced data sets II, Washington, DC*. Citeseer, 2003, pp. 49–56.

[32] G. M. Weiss, “Mining with rarity: a unifying framework,” *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 7–19, 2004.

[33] K. Algra, L. Bouter, A. Hol, J. van Kreveld, D. Andriessen, C. Bijleveld, R. D’Alessandro, J. Dankelman, and P. Werkhoven, “Netherlands code of conduct for research integrity 2018,” 2018.

[34] B. Raskutti and A. Kowalczyk, “Extreme re-balancing for svms: a case study,” *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 60–69, 2004.

[35] N. Japkowicz, “Supervised versus unsupervised binary-learning by feedforward neural networks,” *Machine Learning*, vol. 42, no. 1, pp. 97–122, 2001.

[36] P. Geurts, “Pattern extraction for time series classification,” in *European conference on principles of data mining and knowledge discovery*. Springer, 2001, pp. 115–127.