Utrecht University

MSc-thesis

Game and Media Technology

Modelling appraised interest using eye tracking data

Student: Nikita Iefymov, s/n 5920191

Project supervisors: dr.dr. E.L. van den Broek

dr. M. Zivkovic

Introduction **4**
Baseline literature review **5**
Literature review summary **7**
Visualizations of ETS 8
Influences on ETS 9
Preprocessing of ETS 11
Noise detection and reduction 11
Event detection 12
Feature extraction 13
Feature selection 13
Modelling interest 14
Data exploration **15**
Dataset summary 16
Visualization methods 16
Statistical studies 17
Central Limit Theorem (CLT) analysis 17
Statistical similarity test or Kolmogorov-Smirnov test 17
Implementation methods **19**
Signal processing 19
Preprocessing 20
Denoising 20
Event detection 20
Feature engineering 21
Feature extraction 21
Anomaly detection 22
Feature normalization 23
Feature selection 24
Modelling the user's feedback 24
Classification model of interest from ETS 25
Regression model of interest, complexity and comprehension 25
Classification model of interest from ETS and predicted feedback values 25
Results **26**
Statistical studies 26
Modelling interest from ETS features 28
Description of baseline results 28
Description of results for balanced data 29
Final discussion of the results 29
Modelling interest from textual complexity, comprehension and ETS features 30
Description of results using perceived complexity/comprehension 30
Description of results in predicting complexity/comprehension using regression models 31
Description of results using predicted complexity/comprehension 31
General discussion **32**
Conclusion **33**
References **34**
Appendix A. GUI implementation **36**
Appendix B. Test results **37**
No eye tracking signal 37
Processed dataset 38
Normalized dataset 40
Processed dataset combined with complexity and comprehension 43
Normalized dataset combined with complexity and comprehension 44

Modelling appraised interest using eye tracking data

In this project we use eye tracking data in order to predict implicit user feedback by predicting whether the text was interesting or not. We first study and visualize the data, then apply signal processing methods in order to then extract valuable features and build a classification model. We model appraised interest directly from a set of variables obtained from eye tracking signal. Next, we add to this model predictions of textual complexity and comprehensibility of the text, that are also predicted based on the eye movement features. We show that modelling interest through first predicting textual complexity and comprehension can lead to improved results if carried out properly.

1. Introduction

In the time of information overload, when people read through massive amount of information using digital devices, new research fields emerge in studying user's interaction. Gathering feedback is a crucial part of assessment of any system, and with more advanced systems, gathering feedback can become significantly harder as well. Moreover, the need of user providing explicit feedback using polls and questionnaires is intrusive to the usual user's behaviour and cannot be used at all times, especially in uncontrolled environments. However, implicit feedback is provided naturally in the ways user is interacting, and the challenge is to recognize it and interpret properly. Developing such a system poses a challenge to Affective Computing, as described in [2] by van den Broek. We refer to this study for the general method of capturing emotions using Affective Signal Processing (ASP) approach.

In our study, we attempt to evaluate user's implicit feedback for articles that were read by him or her. In particular, we are interested in evaluating how interesting the text is. In order to facilitate that, we also evaluate perceived textual complexity and level of comprehension of the text. The inspiration for this approach has been taken from the study [1] by van der Sluis et al., where authors explored correlation between perceived interest, complexity and comprehension using text mining methods. In our study, we expand the problem to ASP by using eye tracking signal, recorded by a device while users were reading articles. We apply signal processing techniques in order to process the signal and extract relevant features for the analysis.

The potential application of this research is to develop a closed-loop system, where user's feedback can be evaluated and accounted for naturally. Making the system 'understand' the needs and likes of the user excels human-computer interaction to a whole new level. Gathering and using implicit feedback can be of significant relevance to many use cases, such as news outlets, education, design evaluation and many others. This study shows how eye tracking signal can be used for an unconventional classification process, but also show that the same processing routine can be applied to other signals as well. Despite implementational focus of the project is solely on eye tracking and predicting implicit feedback, we analyze and structure the processing pipeline of the ASP application to the classification problem.

To facilitate the task of modelling interest, we build a full framework for processing and analyzing the data. Although the data, that is used for the project, was acquired prior to the project itself, we emphasise on studying the data and exploring its quality and features. For that, we develop a program with Graphical User Interface (GUI), where the data can be interacted with by means of

visualisation and recording, as well as applying processing steps and evaluating the results right away. We focus on building the full framework for processing the signal in a modular way, where each processing step can be skipped or evaluated separately. In particular, we apply preprocessing methods, detect noise in the signal and look into way of reducing it. Moreover, we classify the events within the eye tracking signal and analyze the anomalies within the signal. Once features of the eye tracking data are identified and their parameters are quantified with statistical values, we produce a dataset that is used for classification and regression models to predict user's feedback.

We will first discuss literature review that was conducted in order to obtain knowledge for conducting the research. We first present the baseline literature review, which is based on the books [3] and 4 by Holmqvist et al., to provide basic knowledge of eye tracking as a field. Next, we give summary of the literature review that was conducted on a list of topics, defined for separate sections of the project. This provides us with understanding of the field and lets us then define the methods that are used, based on the findings from the literature. Method descriptions starts with data exploration methods, where we discuss all the analysis that was conducted before and during the framework development. Implementation methods are then described step by step from preprocessing until the construction of the final model of interest. After the project development is discussed, we present the results, acquired from the described methods. Separate discussions of individual results are then summarized in the general discussion in the end of the thesis.

2. Baseline literature review

When doing literature research to learn about the material required for this project, it was discovered that a handbook [3] by Holmqvist et al provides a well structured and reliable information about a majority of basic knowledge for various fields of research. Hence, this section will focus on providing a summary of all relevant information acquired from the book [3].

First of all, let us define what the eye tracking signal (ETS) is - it is a sequence of gaze points measuring the eye movement. Each sample contains information about a single eye such as position of the eye, pupil size, stimulus that was displayed etc. The samples are taken at a sampling rate with an eye tracking device (eye tracker) turning the eye movement into a digital signal. This signal is what we will further refer to as the raw data. Although majority of eye trackers do include some preprocessing steps before the data can be saved, we still consider the earliest obtainable output as the raw version of the data.

Now we can look into the properties of the signal and what can we learn from it. The most commonly used entry from the raw data is the gaze positions: (x,y) - coordinates on the screen where the person is looking, with which we can analyze eye movement. First event that is recognized from the gaze positions is a fixation. It does not represent eye movement, but on the contrary to an event when the gaze position stays relatively stationary e.g. when being focus on a word and reading it. Such events occur both for texts or other stimuli and are core of how humans use the eyes to perceive information. Average fixation duration is reported to be 200-300 ms. The rapid motion of the eye from one fixation to another (e.g. from word to word) is called a saccade. Saccades are very fast - typically taking 30-80 ms to complete. It should also be noted, that blinks are important events that can happen at any point in ETS resulting in missing or corrupted samples. Depending on the timing of the sampling within the blinking event, the eye tracker may or may not recognize the data entry as a blink. Therefore, additional processing could be required. Smooth pursuit is another type of eye movement, which refers to eyes tracking a moving target, e.g. while watching a video. However, this type of event is not relevant when analyzing eye tracking data acquired during the reading of a text. On the other hand, there are several types of micro-movement that happen within fixations: tremor, microsaccades and drifts. Tremor is a small movement of frequency of around 90 Hz, whose role is unclear, but it is referred to as imprecise muscle control. Drifts are slow eye movements away from the fixation centre, while the role of microsaccades is to quickly bring the eye back to its original

position. These intra-fixational eye movements are mostly studied to understand human neurology and are beyond the scope of this project. Microsaccades are small and fast eye movements that happen during prolonged visual fixations.

The eye tracking device that is used to record the data also influences quality and properties of the data. Besides accuracy and precision of the device, that are tested empirically, a significant device property is the sampling rate (sampling frequency). Devices nowadays cover a spectrum of sampling frequencies from 25 Hz up to 500 Hz and higher. Using Nyquist-Shannon sampling theorem, we can argue that microsaccadic and intra-fixational eye movements with frequencies of at least 150 Hz would require an eye tracking system with sampling rate of detection to be at least 300 Hz. Accuracy is a measure that quantitatively represents how close the recorded samples are to the true values. Whereas precision represents reproducibility of the recording: how close the recorded samples are to each other over the period of time where the true value is constant. Speaking of which, the quality of the data overall and the values of accuracy and precision in particular can be influenced by the experimental setup and other factors during the recording. User's head movement, initial distance between the device and the user or between the device and the screen, lighting - all these factors could cause noise and inaccuracy in the produced signal.

As the authors of [3] point out, data exploration is an important part of the analysis that is often overlooked. The main purpose of it is to get to know the data for further analysis, as well as, check for possible errors and anomalies in the data before they are fed into the data analysis. The goals of data exploration can be summed up into three main ones: check whether data quality is sufficient, look at the distribution of the variables and identify outliers. All of these can be achieved by analyzing various visualizations of the data, such as heat maps of gaze positions, scan path visualization, histograms and box-and-whiskers plots of eye tracking measures etc. It should be noted, that although normal distribution is a regular requirement for statistical tests, eye tracking measures including fixation duration and most saccade measures tend to have skewed distributions, as reported by the authors and other researchers [7,29]. As for identifying the outliers, there are no strict guidelines that are accepted in the field and it leans to case-by-case analysis, where the outliers are attempted to be classified based on their origin. Therefore, the decision on whether to leave them, modify/smooth or drop out also needs to be taken for each case individually. For instance, one of the strategies proposed by Tabachnick and Fidell in [38] and discussed in [3] would be to exclude values that are more than 3.29 standard deviations above or below the mean. Although rare values are not outliers by definition, they may cause undesired effect on the analysis or point to an error of an outside cause.

Filtering and denoising of eye tracking data is an issue that has been becoming addressed more and more within the recent years. This is an essential part of the analysis process as it affects all the subsequent analysis. Some amount of filtering is typically filtered directly by eye tracking device before the signal is being output. However, noise and artefacts still make it through into the signal and they need to be addressed. Optic artefacts are a type of noise among the data samples that derive from recording imperfections due to e.g. eyelash movement, erroneously detect the pupil or corneal reflection. These subconscious movements often appear as sudden spikes in the data and can be identified and removed. A study by Stampe [5] proposes a heuristic filter for detecting and replacing such artifactual samples with neighbouring samples. Another type of noise that can occur in the ETS is low-amplitude high-frequency noise is a cause by impricision of the eye-tracker, as well as the 'jitter' or oculomotor noise of the human eye. However, applying filters to account for that hold the risk of removing authentic eye movements. Lastly, filters can be applying when calculating velocity and acceleration values. If these values are used for event detection algorithm, applying an appropriate filter accommodates a smoother velocity/acceleration signal and easier segmentation of it into the events using thresholding.

Up until this point we have been discussing ETS and its underlying structure consisting of fixations, saccades and other events without going into the details of how those events are detected from the raw data. In general, two common approaches are: identification using dispersion threshold (i-DT) and velocity threshold (i-VT). These two approaches are the opposite of each other in that i-DT

essentially defines fixations as groups of points within the area with allowed dispersion and minimum number of data points, and data entries connecting fixations are considered to be saccades. Meanwhile, i-VT uses velocity of eye-movement at a point of each sample identifying saccades by having velocities above the threshold and saccades having the velocity below it. Dispersion based fixation detection algorithms were presented by multiple researchers, and the algorithm described in the work by Salvucci and Goldberg [6] is used in the project. It should be noted, that velocity and acceleration algorithms are commonly used in packages provided by the eye tracker suppliers, although this class of algorithms typically requires data collected at higher sampling rates (>200Hz) to facilitate precise speed measurements.

3. Literature review summary

In this section we present summaries of literature reviews done on each specific topic. We go into more details regarding the methods we think could be relevant for the project. The scope of our study is quite broad and this section illustrates particularly how many studies are done on just some specific steps from the whole of the processing framework and analysis. The general overview is provided, while for more details on a specific method, please refer to the citations in line.

Here we present a list of 7 topics that were reviewed in a structured literature review. We specify search entries that were used for retrieving relevant papers. Search engine that was used is Google Scholar, and the date of access: 11.07.2018. The period of search from 2011 is used on order to look for more advanced and specific works to build on top of the baseline literature review, since the book that is used for it ([3]) was published in 2011 and summarized the field state at that time.

Topic	Search entry	Period	# results
1. Visualizations of ETS	"eye tracking" AND overview AND "Visualization techniques"	2011-	971
2. Influences on ETS.	"eye tracking" AND calibration AND influence AND "data quality"AND ("sampling frequency" OR "sampling rate")	2011-	362
3. Preprocessing of ETS	"eye tracking" AND ("gap-fill" OR "gap fill-in") AND interpolation	2011-	31
4. Noise detection and reduction	"eye tracking" AND ("noise detection" OR "denoising" OR cleaning OR "noise reduction") AND (technique OR method OR algorithm)	2011-	3800
5. Event detection	"eye tracking" AND ("event detection" OR "fixation detection") AND ("sampling frequency" OR "sampling rate") -"head mounted" -wearable -"smooth pursuit"	2011-	204
6. Feature extraction	"eye tracking" AND "feature extraction"	2011-	4130
7. Anomaly detection	"eye tracking" AND ("anomaly detection" OR "outlier detection")	-	1020

3.1. Visualizations of ETS

To suffice the need of data exploration, the signal needs to be visualized appropriately. Therefore, literature study on visualization techniques of ETS has been conducted. However, the results from the field could not be narrowed down to more than 1000 search result entries without excluding potentially relevant studies. Due to immense saturation of the field, we refer to the survey papers published by Blascheck et al in 2014 [39] and 2017 [8] that summarize the methods used in more than 100 research papers each. Authors classify visualization methods into multiple categories depending on both of the visualization types and stimulus types. The categories are inclusive and the methods are categorized across all of them accordingly.

Visualization-related categories start by the explored dimension being temporal, spatial or a combination of both. Furthermore, visualizations can be animated or static, 2D or 3D and interactive or non-interactive. Besides these self-explanatory categories, authors also distinguish between in-context and not in-context visualizations. In former, stimulus and visualization are linked with each other, such that the data is shown with the context it was gathered in. Lastly, the visualized data can refer to a single user or multiple users.

Stimulus-related categories correlate with major differences in stimuli types. Firstly, stimuli are either static (picture, text etc) or dynamic (video, game etc). Eye tracking data and applied methods differ significantly whether stimulus used is static or dynamic. Next category divides stimuli based on user's interaction into having either active or passive content. Lastly, the content of stimulus can be either 2D or 3D.

Authors describe that the most commonly used visualization techniques for eye tracking data are statistical diagrams such as bar charts, line charts, box plots or scatter plots. Statistical graphics are relevant for descriptive statistics, and therefore, important and widely used for eye tracking data. However, visualization techniques of statistical graphics are typically generic methods that were not specifically designed for eye tracking. While statistical analysis provides quantitative results, visualization techniques allow researchers to consider other aspects of recorded eye tracking data in an exploratory and qualitative way. Visualization techniques help understand spatiotemporal aspects of eye tracking data and complex relationships within the data.

Attention maps are a type of spatial visualization techniques, marking fixation positions as an overlay on a stimulus. This is one of the simplest yet effective strategies applied as early as 1958 by Mackworth and Mackworth. Nowadays, for 2D stimuli the heatmap is the commonly used approach, where the areas of the stimulus are colored differently based on the density of fixations in the region. These visualizations can display Areas of Interest (AOI) to be determined and provide an overview of the dataset overall in a single figure.

Spatio-temporal visualization scanpaths is constructed by reproducing consecutive fixations through saccade lines on the stimulus. A scanpath shows the the order of the events and provide a good basis for the underlying structure analysis. Different approaches exist to highlight fixations, e.g. varying the size of the circles are used to depict them and also convey information on their duration or dispersion. Likewise, saccades can be highlighted differently to add more data to the scanpath e.g. saccade velocity, direction or sequential order (timestamp).

AOI-based visualization techniques employ additional information of the recorded fixation data. AOIs annotate regions or objects of interest on a stimulus. The annotation of AOIs in a static stimulus is often performed by defining bounding shapes around an area, commonly done by automatic fixation clustering algorithms. With information about AOIs, various metrics can be applied to the data depending on the analyst's research questions. Different visualization techniques highlight temporal aspects of the data or relations between AOIs.

3.2. Influences on ETS

The validity of research results based on eye movement analysis are clearly dependent on the quality of eye movement data. Recently, a growing number of researches have been conducted with the goal of formalizing and structuring the knowledge about data quality and potential influences on ETS. When conducting the literature survey, it was found that majority of the recent works quote Holmqvist et al [9] for the industry standard for reporting the data quality. Authors conducted their work in 2012 where they compiled common quality measures, how they are measured, what they reflect and how they can be tested and reproduced in both human experiments and using an artificial eye. Later research in the field mostly focuses on some specific features of data quality and studies it in more depth.

While values of accuracy and precision are generally reported by the manufacturers and the reference values are used by the researchers, recently, some studies have conducted their own evaluations in various experimental setups [11,12]. However, in [9] Holmqvist et al discuss some of the examples how accuracy and precision can influence the recorded data. They discuss how accuracy affects dwell time measures, which is calculated as the total fixation duration within an AOI. This example showed that adding a small offset of 0.5° to the data leads to drastically different measures, where highest dwell time in one of the AOI can reduce by half. Not only this influences the values, but also the distribution of the total dwell time and the measures for the different areas respectively. For the reference, inaccuracy of 0.5° is a value that was reported by most developed eye trackers at that time, that are still used to this date. Two other examples provided in the work, refer to precision and data loss influencing the number and duration of fixations. The data that was recorded from the same eye movement using different precision can vary in its distribution. Inconsistencies in recorded gaze positions lead to detecting short saccades, short fixations or contrary, larger and longer fixations depending on the specific case and the event detection method that is used. Authors show the simulation of adding noise to the data recorded using an artificial eye to explore the consequences in the data and prove the hypothesis.

A number of factors influencing data quality are defined in [9]:

1. Participants can have different eye physiology, varying neurology and psychology. Moreover, wearing glasses or contact lenses can influence the recording as much as having long eyelashes or droopy eyelids. All these factors interfere with the eye image that is processed by the recording device. Blignaut et al study the participants' factor in more detail in [10], where they conduct an experiment with participants from three races, that have different facial features: Asian, Caucasian and African. The study has shown that some factors such as operating distance are equally relevant for all of the participants, while gaze angle, stimulus background color and head movement have led to different trackability,accuracy and/or precision for participants of different races.
2. Operator skill and experience with the device results in recording data with higher quality. The skill set includes adjusting eye to camera angles and mirrors, monitoring the data quality to detect when re-calibration is needed. Moreover, operators can provide clear instructions to the participants and use previous experience in order to eliminate potential problems.
3. The stimulus that is provided for the participant can cause them to behave differently, e.g. move more, blink more often or do an unnatural eye movement pattern. All these factors need to be accounted for when recording and evaluating the data, in order to keep the data quality high.
4. The recording environment is proved to have a strong influence on data quality. One of the most discussed aspects of it being lighting and whether it is high or low, constant or dynamic, natural or artificial etc.
5. The geometric, or experimental setup of relative positions of eye camera, participant and the stimulus has a major influence of the data as well. A number of studies have been conducted

to research experimental setup in more detail, for instance in [11] a mobile eye tracker ‘Eye Tribe’ is evaluated using different setups and then compared to a more advanced eye tracking system. The results of the study show that ‘correct set-up and selection of software to record and process the data are of utmost importance to obtain acceptable results with the low-cost device’ Ooms et al report. Moreover, even if the device is set-up optimally, user’s head movement can influence recorded data as well. Although many manufacturers claim that the devices provide reported accuracy and precision for any accepted head positions, in a recent study [12] Niehorster et al evaluate how different eye trackers perform for unrestrained user behaviour. The following scenarios are tested: rotating head along all three axis, covering an eye or both with patches to study tracking recovering and rotating the head 360° and discovering whether the signal properly discontinues and continues again. The results of the study showed that devices have performed differently, even when values reported by manufacturers were the same. The study was not focused on finding the best device or proving manufacturer numbers wrong, but it showed that further care about the device selection and data quality assumptions should be made in researches where movement in unrestrained, e.g. participant groups of infants.

6. The eye tracking device itself obviously has a large impact on the quality of the recorded data. Many different factors exist in both hardware and software of the eye trackers. Camera resolution, illumination, image analysis algorithms, calibration method etc - all these factors are important and are hard to evaluate separately from the other. In [13] Gibaldi et al evaluate the calibration process of Tobii EyeX eye tracker, by proposing their own calibration process and comparing the resulting data quality. This research shows that improvements could be done even to the steps in the processing pipeline, that are taken care of by the eye tracker. A number of researches have been carried out to compare the performance of different eye trackers, such as [14, 15].

3.3. Preprocessing of ETS

Before analyzing the ETS it is important to make sure that the data quality is sufficient and check if the signal can be curated before it is processed in order to provide better results. In this and following chapters we provide summary of methods applied in the field in order to detect common patterns that change the data and what can be done to restore the signal.

One of the issues with eye tracking data is that in digital measurement systems it is almost unavoidable that some data loss occurs, when a sample cannot be collected for each occasion when the measurement is done. In the context of eye tracking, data loss can be caused by the participant blinking, looking away or when something is put between the eye tracker and the participant [16]. Olsen in [16] reports that these kinds of data losses usually result in larger gaps (>100 ms), whereas shorter gaps can be formed due to abrupt data loss in eye detection and tracking. These types of gaps can cause the signal to result in different outcome of the experiment, report Komogortsev et al in [19]. However, these gaps can be identified and filled-in using present neighbouring samples, in order to restore the lost signal.

Interpolation is a method used widely used in the field, e.g [16,17,18,19] report using linear interpolation in order to fill in detected gaps. The samples before and after the signal was lost are used in order to linearly scale the missing samples. Interpolation is applied for gaps of particular size, typically between one sample and not larger than 100ms. The exceptions to this rule are made in some cases, e.g. in [17] all the gaps were filled in using interpolation regardless of their size, but authors took additional care in order to examine the data to be filled-in correctly and they apply median filter in order to avoid spikes in position changes from the noise.

The other part of preprocessing that is often overlooked and not even reported, is eye selection. The majority of eye trackers are producing binocular data, where the data for both eyes is recorded independently. However, the data can be combined from two signals for each eye into one in different ways. Obvious ways to do it, would be to use data collected from a single chosen eye. Otherwise, the samples can be averaged under different conditions. Special attention is paid to the cases when the data is present only for one eye [16]. Authors, present ‘strict average’, that discards all samples where only one eye has been detected. This approach is also used by [18], since authors report that these samples upon visual inspection were more likely to be inaccurate.

3.4. Noise detection and reduction

Noise in ETS is something that has been studied since first works were published in the field. Previously we generalized the nature of noise and in [3] Homlqvist et al concluded that noise detection and reduction is something that requires case by case nature of solution. However, there are some generalizations that can be made across the field. Therefore, we summarize common techniques in state-of-art research.

In his book [20] Duchowski listed some of the possible causes of noise in the eye tracking data. He proposed that data outside of the given rectangular range of stimulus can be considered noise and eliminated. This also addresses another limitation of eye tracking devices: accuracy degradation in extreme peripheral regions. This approach is used in some researches, such as [34], when data quality suggests that this type of noise is prevalent.

An interesting research [32] done by Medero, where eye tracking data was used in combination with audio data from oral reading were used to predict textual complexity, and more importantly, difficult words and sentences. Medero quotes [33] by Hyrkskykari, where he discusses influence of noise on the eye tracking data. Specifically, the statement that vertical noise is more detrimental to reading analysis than horizontal noise. This refers to AOI-related measures, for example fixations per word, total dwell time per sentence etc. Although horizontal noise may influence the measures,

horizontal noise can cause eye tracking data to switch lines and cause confusion and errors in related measurements. To combat that, [32] and [33] propose identifying line to line saccades by using a sliding window of 6 fixations and identifying a long horizontal saccade towards the beginning of the next line, as well as having at least 2 fixations within a threshold from the right side of the text and at least 3 fixations within a threshold from the left side of the text. Thresholds and window size are to be determined empirically, but this type of measures can be used to access the noise level, as well as, keeping track of where the user is actually reading.

Another type of denoising that is often applied is smoothing. In [16] Olsen introduces two noise reduction functions: moving average and median. Used functions are typically a type of a low pass filter that aims at smoothing out the noise while still preserving the features of the sampled data needed for fixations classification. Another example is presented by Nyström et al in [7], where Savitzky-Golay (SG) smoothing filter is used. This is another type of smoothing filter using moving window that was presented in [21] and is used in the field. Other types of smoothing filters have been developed that are used in eye tracking researches, e.g. [22] introduced a bilateral filtering algorithm written by Ed Vul that was used in [18] by Wass et al.

3.5. Event detection

Event detection is a crucial part of eye tracking data analysis and it is present in the majority of the studies in the field. Moreover, dedicated works have been done to analyze existing methods [6,23] and introduce new improved algorithms[7,26,27]. Below we summarize the state-of-art event detection algorithms and what the optimal conditions for their use are .

The first large family of algorithms use velocity and/or acceleration for event detection. Introduced as early as 1976 by Anliker, velocity was was used to detect saccades, when the velocity within a sample window is above a threshold. However, with the development of both hardware and software for eye-tracking purposes, the data of higher quality became available. Moreover, more research on intrafixational events such as glissades, drift and jitter was done. Velocity-based methods rely on identifying signal at high frequency (mostly >100 Hz [24]) in order to gage velocities and acceleration with higher accuracy. A study [24] was conducted to evaluate influence of sampling frequency on velocity profiles, fixation durations, latency and other measures. Such methods as [16] introduced by Olsen and [19] by Komogortsev et al use same principal of velocity-threshold for event detection, but incorporate it into a more advanced pipeline with preprocessing, noise detection and post-analysis. Moreover, using velocity profiles allows for detection of smaller eye movements such as glissades, as introduced in [7].

The second category of methods of event detection is based on dispersion and dwell-time. Overviewed in [3] and [6], algorithms using Dispersion Threshold (i-DT) and Hidden Markov Model (i-HMM) were evaluated recently in [23]. This approach has seen more recent studies conducted in order to upgrade these event detection algorithms, as summarized by Falkmer et al in [30]. In their work, authors compared and evaluated performance of two dispersion-based algorithms that were using the centroid mode and the start-point mode respectively.

In contrast, area-based algorithms cannot identify fixations at any specific location within stimulus. This fixation-identification method only identifies fixations that occur within specified target areas, i.e., an area-of-interest fixation identification. This method is very basic and was employed back in 1981 by Den Buurman et al in [31], when there was not as much research as there is nowadays. With higher quality of data and more research done in the field, this method is barely used at all. There were found no state-of-art researches that employed AOI-only based fixation detection.

Alternative to all previous techniques, another approach to perform event detection is to use machine learning techniques to classify the events. Recently, there has been more research done using this approach, such as [26] and [27]. In [27] , Zemblys et al present a model that is using a Random Forest (RF) machine machine learning technique for the detection of fixations, saccades and post-saccadic oscillations. The classifier is trained and tested using a vector of features compiled from

the eye tracking data. This research followed their work [26], where similar training data was fed to 10 machine learning algorithms and results were evaluated. The algorithms that authors tested were: K nearest Neighbors (K=3), Linear (LDA) and Quadratic (QDA) Discriminant Analysis, Naive Bayes, SVM with linear and RBF kernels, Decision Tree and RF (32 trees), Ada Boost (with 64 Decision Tree estimators) and Gradient Boosting (with 128 estimators) classifier. The results were compared with specialist data gathered from experts and RF performed the best out of the tested algorithms. The main drawback that Zemblys et al mention is that RF and similar classifiers still require hand crafted features to be extracted first, and postprocessing is required to build meaningful eye-movement events.

3.6. Feature extraction

In this section we provide a summary of different features extracted from the eye tracking data, that are obtained from both raw and aggregated data for further analysis. In section 6.4 we introduce a list of features and measures that we think are of possible use for our case.

In order to quantify ETS using measurements, first events are defined, then their features are evaluated and represented using a number of measurements. The events are defined in the event detection section of the analysis, with fixations, saccades, glissades, smooth pursuits or other events being possible candidates depending on the research and the technique. Features differ for each event, but they can be categorized into movement measures, position measures, count measures and latency and distance measures [3] . For example, fixation count and fixation dispersion are different features of the same event; fixation duration and saccade duration are the same features of different events.

However, even the same features can differ in the way they are defined and how they are described with measures. For example, saccade velocity can be extracted using average velocity throughout the saccade or using peak velocity during the saccade. Although both representing arguably the same feature, they should be analyzed separately and compared to respective values of the same feature. Using the same example of average saccade velocity, we can describe it using only mean value. However, using standard deviation and skewness gives more insight into the distribution of values across the dataset. It was generally found, that mean and standard deviation values are vastly used in the related researches [3,7], but skewness and kurtosis are rarely used and reported.

Some features and their influence have been explored in their own studies. For example, in [32] Medero studies pupil dilation as a predictor of self-explanation. Although authors report that this feature may not be appropriate as the only predictor to differ between two cognitive states, [32] and other sources [3,7] report correlation between pupil dilation and cognitive workload and state. For reference on correlation previously reported by the researches, please refer to sections of [3] and [4] for respective feature and/or measurement.

3.7. Feature selection

After we acquired and analyzed the features of the ETS, we need to feed them to the classifier and obtain the best results possible. In order to achieve greater results, careful feature selection needs to be applied, that determines which features are useful and relevant. Feature selection depends on the model, and more importantly on the problem: supervised vs unsupervised learning. For supervised learning, useful features are the priority, in order to achieve the best accuracy in the testing. However, for unsupervised learning, relevant features that generalize the best are desirable. These generalizations are provided by Guyon and Elisseeff in their work [25], where they describe basics and state-of-art feature selection methods as of 2003. However, their work is proven to be relevant by the recent survey paper [36] by Chandrashekar and Sahin, therefore we will start by generalizing the field as introduced in [25].

Once we established what feature selection is, let's argue for why it is beneficial to use it. Obviously, having a small number of features may not yield better results than having more features. On the other hand, feeding too many features can be as detrimental to the classifier performance. The simplest way to perform feature selection is to rank features based on their usefulness by correlating feature values to the classifier performance, e.g. Pearson correlation coefficient can be used [25]. However, authors argue and exemplify that a variable that is useless on is own can be useful with others. Also, they prove that using variables that are presumably redundant, may reduce noise and consequently provide better classification. And although perfectly correlated variables are truly redundant in the sense that they provide no additional information gain, very highly correlated variables (or anti-correlated) can complement each other. All these statements can be generalized into the need to evaluate features and their combinations at once, rather than apart.

Variable subset selection is an approach in which a subset of variables is selected, that has the best predictive power, rather than ranking individual predictive power of the variables used. Main directions of variable subset selection can be divided into wrappers, filters and embedded methods. Wrappers treat the learning machine as a black box to score subsets of variables in order to achieve the greatest predictive power. Whereas filters select subsets of variables as preprocessing, independently of the chosen predictor. Embedded methods perform variable selection in the process of training and are generally specific to the models. A distinction in methods is made depending on the number of features that are available. For some applications, like text classifications where number of features can be several thousands or even more, exhaustive testing of all possible combination of features is not feasible. In these cases, optimal solution is traded for more time-efficient solutions. Authors of [25] provide examples of computational methods that are used in the field, that belong to the families of methods described above.

A recent feature selection method survey [37] also reports usage of heuristic search algorithms, such as Genetic Algorithms (GA) for finding a subset of features. Although GA was introduced before the research of [37] was conducted, it was not mentioned among feature selection methods. It is one of the possible heuristic based methods, which are essentially another type of wrapper methods, where feature set selection requires a number of computations to be obtained, which is considered its drawback. However, given a small enough set of features, proper wrapper method can yield the best results in combination with a suitable model. Authors of [37] also give state-of-art implementations of feature selection algorithms and compare results of different methods coupled with Support Vector Machine (SVM) and Radial Basis Function Network (BSF). An implementation of a filter method and a GA modification were tested for both SVM and BSF and where SVM obtained comparable accuracy for both feature selection methods, the filter method was outclassed with RBF as the wrapper. Although these results do not reflect advantages of families of methods, authors give practical information about the implementations of the methods and evaluation platform for comparing feature selection techniques.

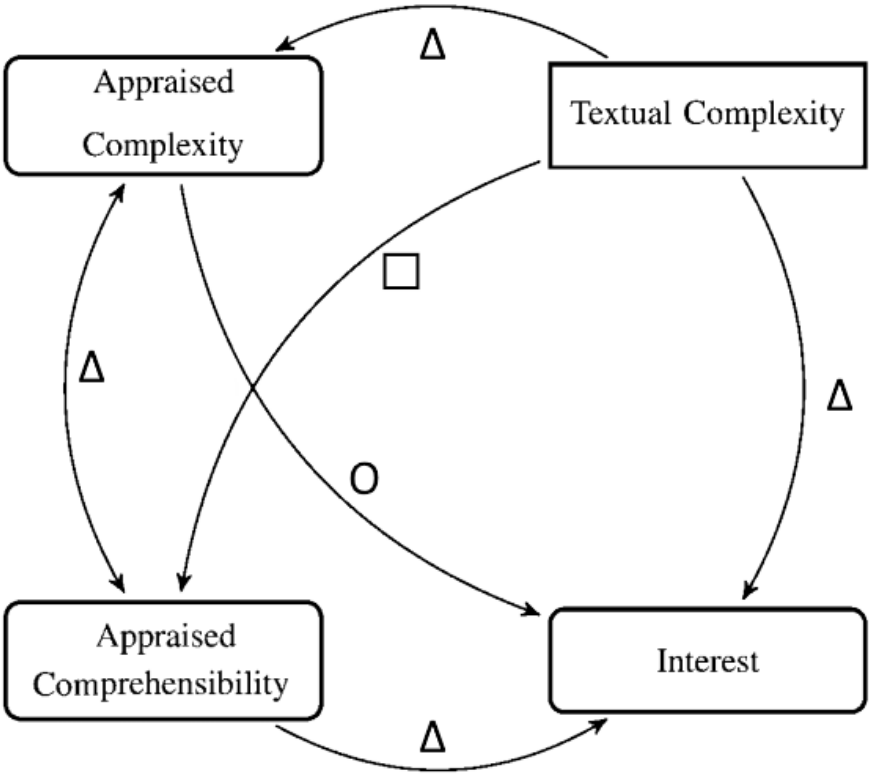
3.8. Modelling interest

All the work that has been done to the ETS signal since the moment it was recorded needs to be finalized by using its final form to predict user's implicit feedback. In order to do that, we model user's interest and build a predictor that could classify a text to be interesting or not. For that we reference [1] by van der Sluis et al and [2] by van den Broek et al in order to tackle this problem of affective computing. In our study we use the same dataset, as van der Sluis et al in [1], where authors predicted text complexity as well as evaluate the influence of the text complexity and perceived complexity to the perceived interest. By using text mining methods, features were extracted to build a regression model to predict complexity. Afterwards, the second study was carried out, where correlations between interest, textual complexity, appraised complexity and appraised comprehensibility were evaluated (see Figure 1). The results of the study report that textual complexity and appraised comprehensibility seemingly captured the influential aspects of appraised

complexity on interest, and it was not a significant determinant for reported interest. Overall, the hypothesis was supported, that more complex stimuli are more interesting, if within the “sweet spot” of being novel-complex, yet comprehensible. The study assesses interest as a measure of perceived relevance of the texts and serves as a good basis for our project to be continued upon. The intention of capturing emotional response and predicting it follows the affective computing principles, and even more so, Affective Signal Processing (ASP) that is described in [2] by van den Broek et al.

Besides providing guidelines and examples on carrying out studies regarding ASP, authors provide general approaches that can be used for classification techniques, applicable in our study. Due to machine learning not being the center focus of our study, we use [2] as a reference for machine learning techniques that can be used as a model for our data. Also, we consult [40] by Holland et al as an eye tracking specific study, that uses a number of machine learning models.

Figure 1. A path diagram, adapted from van der Sluis et al [1]. It shows objective variables (squared boxes) and subjective variables (rounded boxes). Together these explain interest. Legend: O - no significant relation, □ - significant relation, Δ - highly significant relation.



4. Data exploration

In this section we provide information about the dataset we were working with. Next, we discuss methods that were used in order to study the data. It is important to note that development of the framework with GUI (see Appendix A) helped a lot with exploration of the data throughout the project. In the beginning, it facilitated the view of the eye tracking data in the context of visual stimulus. As data processing was advancing further, we could visualize step-by-step improvements due to the integration of methods together into one framework. Besides using the program we developed for studying the data, we also conducted two statistical studies in order to analyze and compare distributions of the recordings that will also be discussed in this section.

4.1. Dataset summary

First, the data set of texts was selected by taking a collection of 14,856 articles from The Guardian. The collection consisted of articles from the following news feeds: culture, environment, financial, market and economics, commentary, life and style, science and technology. To reduce variation caused by differences in article length, all articles were truncated after 1200 characters, followed by three dots to indicate that the story would continue. Next, articles of lower, middle and higher textual complexity were preselected and the final selection of 18 articles was performed based on suitability. The selected news items differed in topics to ensure variation in topical familiarity.

The eye tracking data was recorded while 29 participants were reading a document on the screen. Their average age was 28.60 with standard deviation of 6.06. They voluntarily took part in the experiment. None of the participants was a native English speaker, but all graded their reading literacy as high (mean grade of 4.63, standard deviation of 0.62 within range 1–5, where 5 is the highest). All participants were well-educated; they either had a university degree or were enrolled as a student at a university. A remote SMI RED120 fixed eye tracking device at 60Hz sampling rate was used to track the participants' gaze on a standard TFT monitor with 1280 x 1024 resolution. Software has been used to process and record eye movement signals from the eye tracking device. Viewing was binocular and reported gaze positions are from conjugate gaze: unison gaze with both eyes, meaning the eyes focus in the same direction at the same time. The eye tracking data was analyzed and segmented into following events: saccades, fixation and blinks. The event detection was done with the software provided by the manufacturer. When processing data we do our own event detection after processing the signal, however we do not redo blink detections, since the process was done by the eye tracker and the raw data is not reported prior to blink detection.

After a participant read a text, he or she filled out a questionnaire ranking novelty-complexity, interest and comprehensibility on a Likert scale of 1 to 7. These values are used as perceived novelty and comprehensibility, and the interest estimation that we will try to match in our prediction.

4.2. Visualization methods

In order to study the data we want to visualize the data using methods proposed in the literature. After reading and parsing the data we want to visualize both 'raw data': gaze positions of individual samples and 'aggregated data' of fixations and saccades as classified by the eye tracker. We use a combination of temporal, spatial and spectral visualizations to study different aspects of the data. In order to do that, we developed a Graphic User Interface (GUI) for the project to interact with the data, and after parsing the data, we implemented visualization techniques to inspect the signal. In addition to that, R was used for visualizations of statistical distributions of ETS features.

Firstly, one of the most intuitive visualizations is to reproduce the experiment and see the data in context. For that we visualize the data in 2D space on top of the image of the text, for which the

recording was made, sharing the same local coordinate system in order to see where the gaze positions are located within the text image. However, instead of displaying all samples at once, we can reproduce delays between the samples, since the gap durations are known. Playing back the recording at the same or higher speed gives us the insight into the ordering of the samples and unveils the reading pattern. The same approach was used for both the raw gaze positions data and the aggregated data.

Next, we want to analyze the features of raw ETS, by plotting temporal graphs of x- and y-positions, velocity and acceleration. By looking at how these values are changing over time we can analyze the distributions of the values and detect patterns. We compare plotted data with values expected for the reading pattern recorded for the experiment. Visualizations also provide valuable insight into noise that is present in the data and what preprocessing and denoising needs to be applied in order to account for that.

The same temporal visualizations can be used to gage aggregated data of fixations and saccades. Additionally, we can analyze distributions of key values, such as fixation duration, fixation dispersion and saccade velocity. Further study of statistical features of these distributions can help us evaluate and analyze the data.

4.3. Statistical studies

In this chapter we provide description of the methodology used for conducting two statistical studies. We use this analysis in order to evaluate data distribution and data quality. In particular, the study regarding Central Limit Theorem is conducted for evaluating credibility of the aggregated data reported by the eye tracker. The design choice was made to analyse the raw data ourselves and produce different aggregated data, that could then be compared to the original data. The statistical similarity test is used in order to evaluate correlation between recordings for the same users and same texts. This unveils general correlation in data, but also points out whether normalization may be required in order to compensate for the personal bias. Results of this statistical study may also be used as argumentation for or against personalized models of interest, trained per person. It also shows how well ETS generalizes across different people.

4.3.1. Central Limit Theorem (CLT) analysis

Central Limit Theorem establishes that, in most situations, when values of independent random variables are added, their normalized sum tends towards a normal distribution, even if the variables are not normally distributed. In the case of our data, we want to analyze whether sample means for fixation duration are normally distributed, even if fixation duration is not normally distributed. The fixation duration is chosen to be the test variable as it is as independent random variable in the test context. In case mean fixation duration distribution does not approach normal distribution, it would be an indication of a possible bias or calibration error that have occured in the process of data collection.

Experimental setup:

1. For N, where N is a number of tested files, we randomly draw N files from the dataset.
2. From each file, we randomly draw a set F consisting of S samples.
3. For each set F, we calculate mean fixation duration and store it in a list M
4. After we calculate it for all files, we plot list M as histogram to see the mean fixation distribution
5. We calculate variance of mean fixation durations from values in list M and add it to list V
6. Reset list M and repeat steps 1-5 for all values of N in the test set

7. Plot list V as scatter plot to analyze the change in variance for different N values.

4.3.2. Statistical similarity test or Kolmogorov-Smirnov test

Kolmogorov-Smirnov test (KS test) is a nonparametric test of equality of continuous one-dimensional probability distributions. Two-sample KS test is used to compare two distributions and analyze how they differ. The test results indicate how like is it that two compared samples come from the same distribution.

We use KS test in order to analyze distribution of fixation durations for the same users and the same texts. To do that, we run two experiments, one for users and one for texts. In both cases we use two-sample KS test to pairwise analyze all the recordings for each user for the first study, or each text for the second study. The results of KS tests are used to determine influence of user and text features on fixation duration and ETS features in general. The results of statistical similarity test are important for identifying influences on the ETS and how the normalization should be performed.

Experimental setup:
Test A: per user analysis

1. Define collections C_1,C_2...C_n, where each collection consists of all samples from a single file that contains recording for the user.
2. For each pair i and j, where 1 <= i <=n and 1 <= j <= n, perform two-sample Kolmogorov-Smirnov test for collections C_i and C_j. Store p-value and d-statistic at (i,j) position in respective matrices P and D.
3. After all the pairs of collections are evaluated, the matrices P and D are complete and can be printed to have complete set of evaluations.
4. Reset matrices P and D and repeat steps 1-3 for all users.

Test B: per text analysis

1. Define collections C_1,C_2...C_n, where each collection consists of all samples from a single file that contains recording for the text.
2. For each pair i and j, where 1 <= i <=n and 1 <= j <= n, perform two-sample Kolmogorov-Smirnov test for collections C_i and C_j. Store p-value and d-statistic at (i,j) position in respective matrices P and D.
3. After all the pairs of collections are evaluated, the matrices P and D are complete and can be printed to have complete set of evaluations.
4. Reset matrices P and D and repeat steps 1-3 for all texts.

5. Implementation methods

In this section we discuss implementation methods that are used at different stages of the project. In order to provide a general overview of the full ensemble of methods, we present the processing pipeline, and then discuss every step of the process in detail. Signal processing module takes as raw data as an input, provided by the eye tracker as it is. After the outlier detection, interpolation and filtering are applied, the event detection is run to obtain processed aggregated data. It should be noted, that all processing steps are implemented in the same framework with GUI (see Appendix A) in a modular fashion. This means, that we can try and test different combinations of processing steps, as well as different parameter values, where those apply. In this section we go into more details on the implementation of each particular module with the design choices that were made and the parameters that were chosen.

After signal processing is concluded and aggregated data is produced, we proceed with feature engineering to calculation of measurements that describe the ETS. After feature extraction, normalization and selection, we use these variables for machine learning models in order to predict user's feedback.

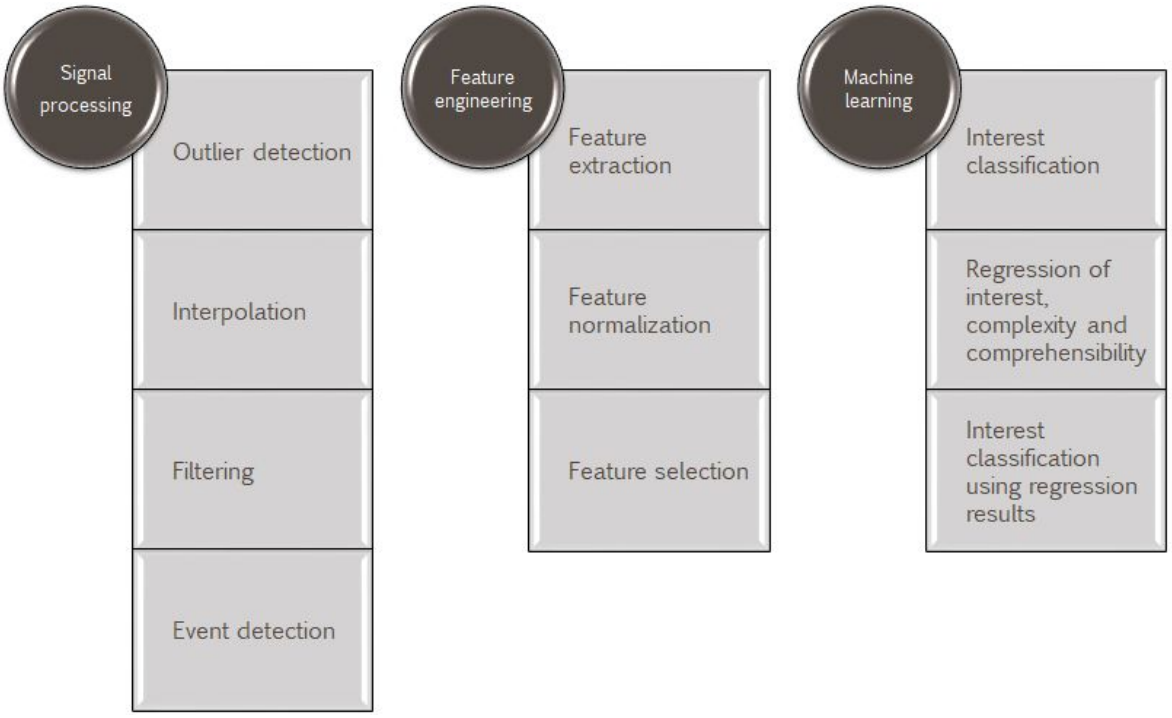


Figure 2. The structure of implemented methods

5.1. Signal processing

In this chapter we discuss steps of signal processing that we use for processing the data. All of the following steps were implemented in the same framework, that is described in Appendix A. We start by taking raw data, containing gaze positions, pupil dilation and timestamps. We apply preprocessing of removing outliers and filling-in the gaps, apply smoothing filter to denoise the data and then we aggregate the data to produce a dataset containing fixations and saccades. This dataset of 'processed data' is then used throughout the project as the aggregated data acquired from the ETS.

5.1.1. Preprocessing

After the signal is acquired and studied, the processing pipeline starts with preprocessing. We want to make sure that relevant and proper data is passed along the pipeline. In order to achieve that, we apply techniques discovered from the literature review.

Since the texts that were read by the participants were all located at the same place, we can define a bounding box, within which all samples correspond to a user reading the text, while samples outside of the bounding box can be discarded [20,34]. Samples found outside of the bounding box were observed at the start and the end of the recording, possibly before and after the user was reading the text. Also, throughout the reading process, there were samples that occured at random positions outside of the text, which we treat as noise, as majority of the samples do not even comprise full fixations that could be classified as user distraction or loss of focus.

Next up, we identify gaps where data is missing and which should be filled in. In order to do that, we are looking for pairs of samples that are more than x milliseconds apart, where x is the minimum gap size. Values of x being 100ms and 150ms are tested, as those values are reported in [16] and [18]. These gaps are filled in using linear interpolation by determining the scaling factor based on the gap duration and interpolating the missing samples using formula introduced by Olsen in [16].

Since the data we are using provides conjugate gaze data, the positions of unison gaze with both eyes are reported. Therefore there is no need for implementation of eye selection algorithm, when reported gaze positions of both eyes are identical.

5.1.2. Denoising

After the signal is preprocessed, we apply noise detection and reduction techniques. The first denoising step has already being explained, as it involves discarding data which does not relate to the experiment itself: samples outside of the text image. Next, we want to apply a low-pass or a smoothing filter in order to reduce high frequency noise. By applying Shannon-Hartley theorem, we can identify that the highest frequency signal that can be captured without aliasing at 30 Hz for sampling rate of 60 Hz. However, the noise that is introduced in ETS due to jitter is reported to have a frequency of around 150 Hz or higher [4]. Therefore, we can only use a smoothing filter in order to combat this, since the signal cannot be excluded explicitly.

We apply Savitzky-Golay filter [21] of window size 5 and filter order 2 in order to smooth the data, as proposed and used in [7]. Although authors of [7] base filter length on minimum saccade duration of 20 ms, due to low sampling rage, we use the minimum saccade duration of 50 ms, or roughly 2.5 samples. Original value of 20 ms is equivalent to slightly more than a single sample, and we need to provide window size that is large enough for smoothing, while retaining the nature of the signal. Additional denoising is applied after the anomaly detection is performed with the extracted features of the ETS, that will be discussed in Section 6.5.

5.1.3. Event detection

After preprocessing and denoising we acquire ‘clean’ signal comprising of gaze positions that can be used for event detection, in order to identify fixations and saccades. The dataset that was provided already had labeling from event detection algorithm applied in the eye tracking software, however we do our own event detections after manipulating the data and obtaining a ‘clean’ signal. Moreover, the original events reported by the eye tracker can be used as a reference in order to evaluate the performance of implemented event detection.

In order to choose an event detection algorithm to apply, we used key data features. First of all, presented stimuli were all texts, static images and we are interested in fixations and saccades. Secondly, low sampling frequency does not allow for detection of itrafixational events such as glissades and jitter, as explained in the literature summary.

A choice was made in favor of a dispersion-based event detection algorithm, since it should perform well for the given data, as reported by Salvucci and Goldberg in [6]. Although more recent velocity-based algorithms were introduced, we believe that the quality of data is not sufficient for modern sophisticated i-VT algorithms. The implementation of i-DT from *emov* R package is used, with parameters being: maximal dispersion of 30 px (or about 0.5° for the experimental setup) and minimal fixation duration of 6 samples (or ~100ms at 60 Hz). These parameters are applied based on recommendations from [6] and are adapted for the experimental setup.

5.2. Feature engineering

Although the resulting signal may seem high quality and relevance, we need to define a set of features and their parameters that define and describe the signal. Moreover, analysis of these features may be used for anomaly detection within the dataset. Some of the features are reported by related literature, so we can compare our findings. Otherwise, in order to adequately analyze the data ourselves, but also to produce comparable results that can be related to, we apply feature normalizations to construct a separate dataset where variables of akk features are normalized per participant.

5.2.1. Feature extraction

After the events are defined within the ETS, we need to define the features that describe valuable properties of it. In this section we describe features that we extract from the events, due to their potential significance in predicting the interest in the end model. The literature survey described in Section 4.6 covers works, such as [3] and [4] that mention reported correlations of these features in other researches.

Some features are common for both fixations and saccades, such as their count and duration. Count is measured in the total number and rate per second, with the latter being a normalized value. In addition, we measure the mean number of fixations per line, as it has been reported to be a potential indicator of the text comprehensibility or complexity [4]. Another common feature for both fixations and saccades in the duration and its distribution has been researched in a number of researches [3,4,17]. Measurements, such as the mean duration, duration standard deviation, variance and skewness are used in these works. We also add kurtosis as an additional representation of the distribution of the duration. All these statistical measures are commonly used for assessing the raw data quality [9] and the event detection quality [29]. These measures are also used for the final model, e.g. [40] by Holland et al. An additional measure for the fixation duration is the mean reading time per line, as proposed and used in [4] and [17].

We extract the same set of statistical measurements for distributions of fixation dispersion, saccade amplitude, saccade velocity and saccade acceleration. Although not all of these features

have been only quantified with mean and standard deviation values in the literature, we believe that additional feature statistics can be of use for the machine learning models. Fixation duration is calculated as a radius of the fixation, measured in º. Saccade amplitude is first calculated in pixels on screen and then in º, since the distance between the participant and the screen is known. Saccade velocity can be described by two different values: mean saccade velocity and the peak saccade velocity throughout the saccade duration. We use statistics for distributions of mean saccade velocity, measured at º/s. Peak saccade velocity and likewise, mean and peak saccade acceleration cannot be adequately evaluated due to the low sampling frequency and therefore excluded. Namely, a saccade can be represented with no samples, when it takes place between two fixations in less than 1/60 s. This leads to inability to evaluate peak velocity and acceleration throughout the saccade.

To complement statistical measurements, we use other features of the eye tracking data, that are extracted using the data context. In particular, regressions (or backward saccades) are reported to be correlated with the textual complexity [42]. Regressions are defined as saccades with amplitude larger than 2º [4] directed towards the part of the text, that was already read. Regression count is measured by the total number of regressions, their rate per second and the mean number of regressions per line. Also, measures of pupil dilation are represented with mean pupil dilation and Pupillary Unrest Index, introduced in [41] and argued to be a reliable pupil measure. This feature might be useful, due to known correlation between cognitive workload and pupil dilation [4].

Table 1. All the measurements which are calculated from the features of the ETS. The table defines event and feature that the measure describes and whether or not it is normalized. Also, we provide references to the related literature. Below the table we define what comprises ***Distribution metrics.***

Feature	Measure	Statistic	norm.	ref.
Fixation	Count	Number	✘	[3, 7]
		Rate	✔	[3, 7]
		Mean number per line	✘	[3, 7]
	Duration	***Distribution metrics***	✔	[3, 7]
		Mean reading time per line	✔	[4,32]
	Dispersion	***Distribution metrics***	✔	[3, 7]
Saccade	Count	Number	✘	[3, 7]
		Rate	✔	[3, 7]
	Regression count	Number	✘	[3, 7]
		Rate	✔	[3, 7]
		Mean number per line	✘	[4,32]
	Duration	***Distribution metrics***	✔	[3, 7]
	Amplitude	***Distribution metrics***	✔	[3, 7]
	Mean velocity	***Distribution metrics***	✔	[3, 7]
Scanpath	Length	Length	✘	[4,36]
	Duration	Total reading time	✘	[3, 7]
Pupil	Dilation	Mean	✔	[3, 7]
		Pupillary Unrest Index	✔	[4, 41]

Distribution metrics:	Mean	✔	[3, 7]
	Standard deviation	✔	[3, 7]
	Variance	✔	[3, 7]
	Skewness	✔	[4, 17]
	Kurtosis	✔	
	Mean reading time per line	✔	[4,32]

5.2.2. Anomaly detection

It is important to ensure that the data is adequate and relevant early on before processing it, as well as making sure that any other anomalies that can occur later down the pipeline can be caught, studied and taken care of. Therefore, we describe our strategy of anomaly detection that we first apply at raw data level and after event detection.

When analyzing raw data, we only want to keep the samples that refer to reading the text. Therefore, we are looking to exclude data that is either noise and has random nature, or samples that were simply taken when not reading the text or refer to person being distracted. Also, we want to exclude recordings that do not carry enough adequate data. However, given a large dataset, we need to automate this process after studying the data tendencies. After analyzing the dataset by visualizing gaze positions on top of the text images, different classes of outliers were identified:

- Recordings with large data loss - recordings with less than 1000 samples were discarded, since average number of samples is more than 30000 samples for a fully read article.
- Recordings with samples not following the pattern of reading the text - recordings with less than 10 line-to-line saccades are discarded, since all displayed articles had at least 12 lines in them.
- Recordings with an offset - large offset is detected with the number of line-to-line saccades, since it heavily affects the samples near the edges.

Therefore, for the first class of abnormal recordings can be classified by calculating amount of data loss: total gap duration, a number of missing samples and proportion of samples missing. Based on these values, we can define a threshold after which the recordings can be considered having too little data.

For the second class, we mean that a large portion of gaze positions do not follow the text along, but the eye movement is recorded to happen in a random fashion, no matter within the text image, or outside of it. The issue with this class, is that all the samples may seem to have quite similar statistical properties regarding their positions, but the pattern of reading is not apparent. Therefore, fixations and saccades that are identified from this type of dataset will not make sense and lead to meaningful results in the analysis. This class is extremely difficult to identify automatically based on some features, so we developed a measure of a number of line-to-line saccades. Namely, these are long horizontal saccades starting at the end of a line and ending at the start of the next line. We used the implementation method introduced in [32] by Medero, where line-to-line saccades are identified over a window size of 6 fixations, where at least 2 fixations are located at the end of the previous line, and at least 3 fixations are located and the beginning of the next line. The end and the beginning of a line are defined by the distance to the edge of the displayed text. These saccades were found only in the recordings where data points did follow the text reading pattern. In case the number of saccades is much lower than number of lines in the displayed text, we can consider this text to not follow the pattern. Certain properties of the scanpath could be used for this classification as well, but due to inconsistencies in the data, we were not able to identify a feature that would be a great indicator.

Another type of anomalies, that happens in the eye tracking data is caused by the presence of an offset. This happens due to poor calibration or unexpected head movement that was not accounted for by the processing algorithms in the eye tracker software. Recordings with offset have a consistent spacing between the actual gaze position and the recorded samples. For example, due to an offset, one of the corners of the text is never read, but instead all the samples on the side of the corner are shifted by this offset. Nevertheless, if the reading pattern is still clear, we do accept this type of data if it will pass tests for the previous types of anomalies. Offset would be detrimental to features based on Area Of Interest (AOI) and positional features, but we are not using them in this project.

5.2.3. Feature normalization

In order to produce commensurate quantifications of the features, that are extracted from the signal, we need to normalize the values to exclude influence of personal and textual features on the measurements. For adequate analysis of the values, that are recorded for different people, we need to normalize measures, that are collected each single participant. For example, if a person takes on average significantly longer to read the text, we should account for that, so that it can be compared with a recording for a person who reads significantly faster. If normalization is done adequately, the same patterns can be found in recordings for both of these participants. On the other hand, normalization of all values, that are collected during the experiments, allows for a more appropriate comparison of these values, to values that are obtained using different experimental setup, e.g. stimulus, screen etc.

In Table 1 we noted which values are normalized once they are calculated, meaning they are commensurate within the dataset. However, all these values are, to different extent, influenced by the personal features and the reading pattern. Therefore, all the values are normalized within their distributions per person. We refer to common normalization methods, as summarized by van den Broek in [2], where normalization for human biosignal was proposed and argued for. We apply baseline correction, also known as standardization, where a value is replace with a difference between the value and the mean of distribution, divided by the standard deviation of the distribution. This method is not too sensitive to outliers, suits the purpose and easy to reproduce and relate to.

5.2.4. Feature selection

Previously, we described and listed 38 statistical measurements that are calculated from event and their features of the ETS, next we need to narrow down which of these measurements, or in the context of machine learning - named features or variables, will be selected and used for modelling the interest. In order to do that, we apply feature selection method to select a set of features that results in the best performance for the final model. After analyzing the related literature, summary for which we provided in Section 4.7, we chose the methods we use for this project.

One of the feature selection methods of choice is the implementation from *EFS* R package, introduced in [43] by Neumann et al. Authors proposed and developed a normalized quantitative score of all relevant features by using multiple techniques. The following values are averaged and the mean is taken as the score:

- Median: p-values from Wilcoxon signed-rank test;
- Spearman's rank correlation test;
- Pearson's product moment correlation test;
- Beta-Values of logistic regression.

  For more details on each score refer to [43] on specifics of implementation.

We use this score in order to rank the variables and select a set of features above a certain threshold of correlation.

Another feature selection method that we use is embedded in RF model. Authors of [43] also introduce using RF as a feature selection method, but we use the model for classification, as well as feature selection, as it will be discussed below. However, we can also use the feature selection of RF for the second model that we use for predicting the interest. For example, R package *randomForest* provides cross-validation feature selection algorithms that evaluates model performance using different number of features and different sets of features. This can be used both for producing the final set of features and for evaluating the influence of the number of variables on the performance of the classifier.

As we previously summarized in the literature survey on feature selection, a number of techniques of different kind can be applied. As such, we use *FSelector* package from R to evaluate

the performance of search algorithms: best first search, greedy search and exhaustive search. Combining multiple feature selection techniques can yield the best result, and applying search algorithms on a reduced set of features is a common way to determine the optimal set of features for the classification process.

5.3. Modelling the user's feedback

The ultimate goal of our research is to model interest and predict it using measurements obtained from the eye tracking data. In order to do so, machine learning classification model is used to predict whether a text is interesting, based on a vector consisting of a number of measurements that were preselected. We conduct two studies for predicting interest, where in the first study we estimate it directly from all the measures, while in the second study we apply a model proposed by van der Sluis et al in [1], where we predict complexity and comprehensibility and use obtained values to model the interest, alongside the ETS features.

Due to Machine Learning (ML) not being the primary focus of our study, we use several commonly used approaches to model the interest, with implementations from R libraries. The selection of ML models was based on literature research [3,1,2] and we chose SVM and RF to be used. SVM is a commonly used classifier that has shown both good performance and decent interpretability of the results. Although the goal of the research is to build a model with the highest success rate in testing, we believe that understanding the impact of different eye tracking measurements can be helpful for further research on this topic. For similar reasons, RF model has been chosen, that on top of performance and transparency, provides embedded feature selection. This lets us compare the feature selection that is performed by search methods and the results of embedded feature selection. In all cases, models are tested using 10-fold Cross-Validation (CV) executed 10 times, leading to a total of 100 tests per model. This is done in order to produce consistent results that do not fluctuate depending on the sampling process.

We use SVM implementation from *e1071* R package with radial kernel function, as it showed the best results in preliminary testing, as well as optimization of the parameters using *tune.svm()* function from *e1071* package in R. This method allows us to use grid search in order to find the best performing parameters for SVM. As for RF, we use the implementation from R package *randomForest* with number of grown trees being 500.

5.3.1. Classification model of interest from ETS

In our first study, we use a vector of feature parameters comprised of all of the measurements, as described in 6.4. Then, this vector is reduced using feature selection method. The same approach is used for training both SVM and RF models. In this study we aim to evaluate how well can the perceived interest of a text be predicted from eye movements during the reading.

Since we treat interest classification as a binary classification problem, balance between classes is important. Therefore we also analyze the influence of how we label the data to be 'interesting' or 'not interesting' as a balancing method. To do so we modify a 'Threshold of interest' t_i - a grade, above which all grades of interest classify respective texts as interesting. To add onto that, we apply resampling methods in order to artificially balance the data and model the training set with balanced distribution of the classes. More details and results will be discussed in the following section.

5.3.2. Regression model of interest, complexity and comprehension

To model the interest through complexity and comprehension, as proposed by van der Sluis et al in [1], we need to predict the values of these feedback values from the training set. In the dataset,

all feedback values were evaluated on the 1 to 7 semantic-differential scale. We use these values to train a regression model to predict one of these values and analyze the performance of the model. We evaluate performance of the regression models using Mean Squared Error (MSE) and relative accuracy, standing for the percentage of predictions that can be considered accurate. Considering the grade scale being from 1 through 7, if the distance between the predicted and true value is less or equal to 1, we consider prediction to be accurate.

5.3.3. Classification model of interest from ETS and predicted feedback values

In the second study, we train additional models for comprehensibility and complexity of the text, using the same approach, as in the first study. We combine all of the original measurements for predicting the interest with the predicted complexity and comprehensibility of the text. We employ the approach proposed by van der Sluis et al in [1] and apply it to the eye tracking data. This study lets us evaluate the performance of this advanced model of interest with the results obtained from the previous study and get an insight into the correlation between interest and comprehensibility-complexity. To add to that, we evaluate performance of the classification model with and without ETS features, to evaluate whether eye tracking data does provide valuable insight to the classification process. We also compare results achieved when using grades provided by the users (true values), and grades predicted by the trained regression models (predicted values).

6. Results

In this section we discuss results obtained from implementing methods, that were just described. We provide short context for the experimental setup and parameters, that were not previously defined in the method section. After that, we use results that are indicative and relevant in our opinion and provide brief interpretation of the results. For tables of complete results, refer to Appendix B.

6.1. Statistical studies

Two statistical studies were first conducted before the implementation of the processing framework started. The goal was to learn more about the data distribution and the patterns within the dataset. Also, during the analysis process, we could gage the quality of the data when processing it. After the processing pipeline was implemented and we acquired what we consider to be a clean, processed signal, we repeated both statistical studies to evaluate whether the outcome changed. Therefore, in this chapter we present results for both the original dataset - aggregated data of fixations and saccades reported by the eye tracker, and the processed dataset - aggregated data acquired from preprocessed raw data, using the developed processing framework.

Results of CLT-test

We test Central Limit Theorem in order to analyze whether its hypothesis is met for eye tracking data, that we use for the project. In particular, we analyze fixation duration distribution from both the original aggregated data, reported by the eye tracker, and the aggregated data produced by the signal processing pipeline. Although fixation duration itself is not normally distributed [3,7], aggregated data, such as mean fixation duration should follow normal distribution. In Figures 3 and 4 we can see the histograms of mean fixation durations for original and processed aggregated data respectively. Mean fixation duration for aggregated data reported by the eye tracker is distributed in a quadratic or

triangle shape. This is caused by the wide range of values and the bins containing outliers are more populated than the normal distribution suggests. However, distribution of mean fixation durations for processed data tends towards normal distribution. Although notably fixations have minimum duration threshold in the event detection. The difference in distributions may be caused by the outlier and anomaly detection that we apply. Therefore, the resulting data is significantly more consistent and follows our expectations.

Figure 3. Histogram of mean fixation duration in the original aggregated data reported by the eye tracker. Number of bins = 8

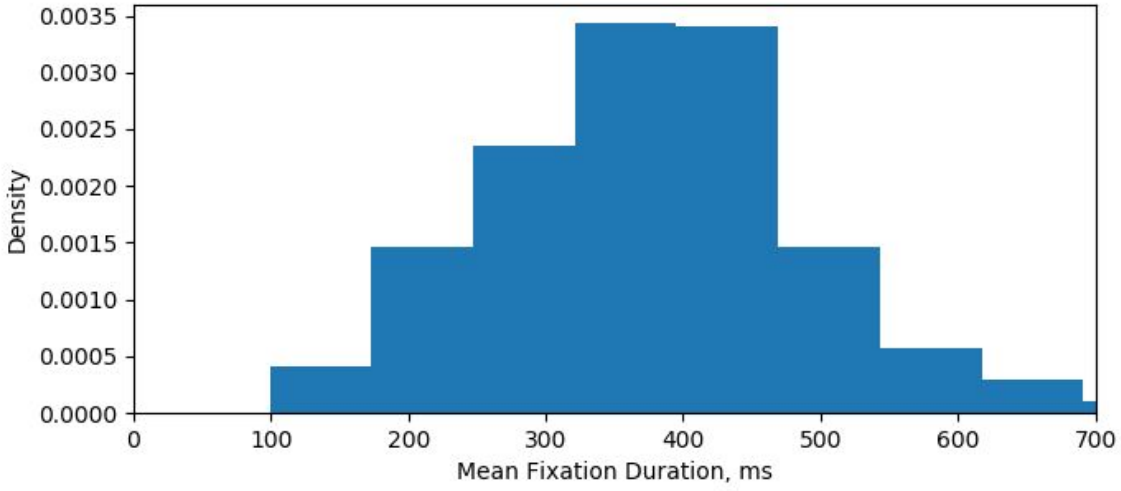
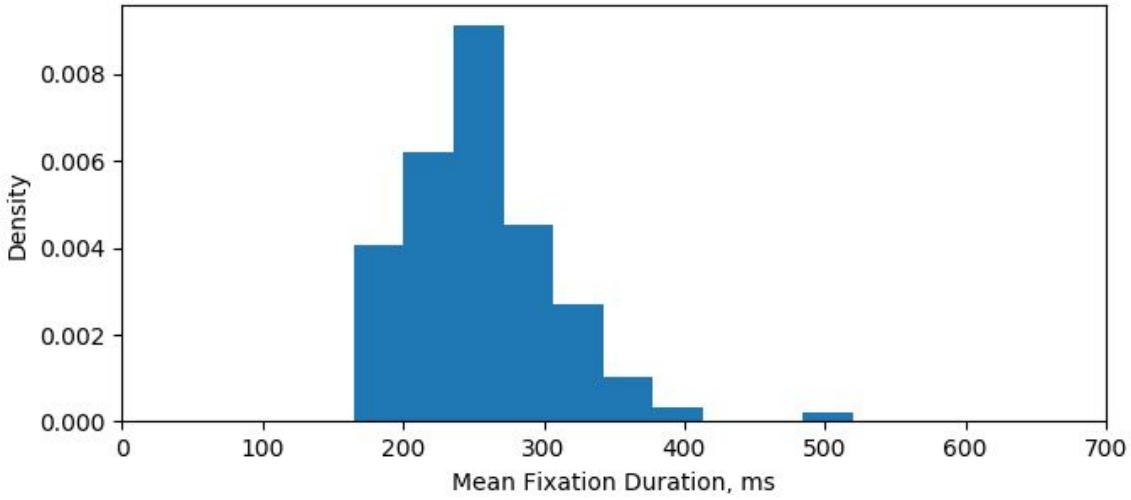


Figure 4. Histogram of mean fixation duration in the processed aggregated data produced by the signal processing pipeline. Number of bins = 8



Results of KS-test

The statistical similarity test, in particular Kolmogorov-Smirnov (KS) test, was used to compare distributions of fixation durations. All of the recordings for the same text or user were compared in pairs and the test results describe whether the data belongs to the same distribution.

H_0-hypothesis: two samples belong to the same distribution

H_1-hypothesis: two samples do not belong to the same distribution

For significance level α, null hypothesis is rejected when $p < \alpha$. The tests were conducted for all texts and users using both the original aggregated data, as reported by the eye tracker, and the processed data, obtained using full pipeline, as implemented for the project. Table 2 shows the results by summarizing it with the total percentage of the pairs, for which the null hypothesis is rejected at $\alpha = 0.05$.

Table 2. Percentage of pairs with $p < \alpha$ at $\alpha = 0.05$.

Dataset	Texts	Users
Original data	87.07%	37.57%
Processed data	69.14%	25.00%

Overall, no significant correlation between recordings for the same texts was found. It was expected, as all the displayed texts were truncated to be the same length and overall as visual stimuli were extremely similar. However, the rejection rate is more than twice lower for the tests done per user. At 37.5% for the original data and 25% for the processed data, we can make a strong assumption that there is stronger correlation between recordings done for the same participants. This is supported by the observations from the data visualization, as data quality was influenced by the experimental setup for each participant. This also caused data from a number of participants to be excluded from the final data selection.

Results of KS-test also illustrate the general influence of pre- and post-processing on the data. The first cause for higher correlation in data is excluding of the outliers both within and between recordings. The second reason, is the smoothing, filtering and denoising that make for more consistent data, hence resulting in higher correlation in such features as fixation duration, that was compared in this statistical study.

6.2. Modelling interest from ETS features

In this chapter we discuss results achieved in modelling interest using features extracted from the eye tracking data. Two models: Random Forest (RF) and Support Vector Machine (SVM) were used, as described previously in Section 5.3. Tests were conducted on two datasets: the processed dataset constructed from applying full signal processing pipeline and calculating all the statistical measures, and the same dataset, where all measurements are normalized per participant. We call these two sets ‘processed’ and ‘normalized’ respectively. Some of the parameters were defined in preliminary testing: e.g. number of trees in RF, SVM parameter grid search parameter tuning, etc. On the other hand, a number of testing parameters were used in the testing process in order to evaluate their impact and find the best set of testing parameters. Refer to Appendix B for the original tables of the test results.

6.2.1. Description of baseline results

Results for both datasets using both models, feature filtering, feature selection and scoring functions were obtained and evaluated. Test results are reported with 5 values, averaged across all the tests performed for the experimental setup. Training and testing was performed using 10-fold cross validation and repeating the process 10 times. Therefore, overall 100 tests were done to evaluate each set of parameters in order to produce results that are consistent over time. When using smaller number of tests, the mean accuracy of a model could fluctuate by more than 1%, while 100

was found to produce results that are comparable, with the mean difference in accuracy of less than 1%.

When evaluating performance of the models when using two different functions, it was found that accuracy led to higher or equal resulting accuracy in overwhelming portion of the tests (all but one instance). Although using Negative Predictive Value as the scoring function resulted in more instances of correctly classified negative values, the trade-off of False Negatives influenced the accuracy too much, and the accuracy did not recover.

Results for both feature selection methods: best-first search and forward-search were identical in the majority of the cases due to a very similar approach. Therefore, the use of just one of these for further tests is possible. Due to high computational demands of backward search and exhaustive search, these techniques can only be applied on the most promising test setups and with the reduced feature space.

Regarding reducing the feature space, applying feature filtering based on 8 ensembled relevancy rankings performed reasonably well, leading to a drop in performance for some test setups. However, the best performing setups with the original feature space showed comparable results with the reduced feature space too. In one such case, reducing the feature space led to improved results even for forward-search. This suggests that using this method of feature filtering can be applied for some of the setups in order to reduce computational complexity and allow more efficient feature selection methods to be applied.

As for the results for different models, Random Forest outperformed both linear SVM and radial SVM in the majority of the tests. Moreover, SVM suffered more from the imbalance in the data. All the tests with linear SVM lead to 100% prediction rate of the prevalent class of interesting articles. Therefore, the accuracy of linear SVM was equal to the True Positive Rate and portion of positives in the input data. Therefore, linear SVM was excluded from further tests. However, SVM with radial kernel function showed similar behaviour to RF. It is unclear whether it generalizes minor class better or worse than RF, due to mixed results. However, performance of the RF and radial SVM should be evaluated further with more advanced experimental setups.

6.2.2. Description of results for balanced data

Another domain of interest in the results is the influence of imbalance in the data. Namely, the fact that class of interesting articles comprises above 80% of the dataset. This leads to higher accuracy trade-off for the models that aim to identify both classes, rather than ‘follow the trend’ of mostly modelling the prevalent class. The problem of imbalanced data is discussed in [44,45,46] and possible solutions are proposed. Due to machine learning not being the primary focus of the study, we did not apply advanced strategies, such as boosting, cost-sensitive learning or adapting existing machine learning methods to account for data imbalance [44]. We tackle data imbalance issue on the data-level, firstly by using a different ‘threshold’ for the interest grades, that determines what user’s feedback can label the article to be interesting or not interesting. Besides that, we use resampling methods to model the distribution of the imbalanced dataset by undersampling the prevalent class, oversampling the minor class or combination of both. For that we an established technique ROSE, a successor of SMOTE[45,46,47], to produce balanced dataset and use it for training and testing of the model.

Table 3. The highest classification accuracy obtained for the original two datasets and their resampled versions using ROSE, using different threshold of interest t_i.

Dataset	Results for t_i=3.5	Results for t_i=4	Results for t_i=4.5
Processed	83.51%	73.48%	66.29%
Normalized	83.67%	70.78%	65.64%
ROSE processed	74.84%	65.39%	62.85%
ROSE normalized	70.26%	64.22%	59.48%

6.2.3. Final discussion of the results

We tested a number of classification models that aim to identify whether text was interesting or not, based on the eye tracking measurements. Overall, results are promising and they show that eye movement features can be used for such prediction of implicit user feedback. The average accuracy achieved by 10-fold cross validation repeated 10 times was up to 85% We also discussed the problem of imbalance distribution in class labels and how it can be approached. Although there was a drop in performance when applying majority of the balancing methods, different methods could provide better results. We also achieved 100% accuracy when using ROSE re-sampling method on the full dataset and using resampled data for cross-validation. Although, we believe that perfect result is caused by synthetic data generation of the method, it is indicative that if the measures had similar distributions with larger sample size and balanced ratio between class sizes, high accuracy can be achieved.

After analyzing of all the sets of input variables, we can identify most promising test setups for our next study and for the future work as well. Namely, best-first search and exhaustive search on a reduced feature space, with both searches using accuracy as scoring function. We propose using ensemble of feature selection methods to evaluate relevancy of all features and use resulting normalized importance values for feature filtering and reducing feature space[48].

6.3. Modelling interest from textual complexity, comprehension and ETS features

In the second study, we investigated how perceived interest can be modeled from eye tracking measures incorporated into the model proposed by van der Sluis et al in [1], where correlation between interest, textual complexity and comprehension was studied. The main premise of this study is namely: interest can be predicted with higher accuracy if perception of textual complexity and comprehension is known. In order to test this premise, we add the grades for complexity and comprehension provided by the users as another input variable and evaluate the performance of the resulting models. Then we evaluate to what extent ETS predicts interest, complexity and comprehension by analyzing performance of regression models that predict respective scores. And finally we test whether using predicted feedback values improves interest classification model. It should be noted, that we also use reduced set of testing parameters, based on the conclusions made from the previous study, in order to focus on more optimal setups.

6.3.1. Description of results using perceived complexity/comprehension

In table 4 we present the best results achieved for data with and without normalization from the first study and compare them with the best results when adding as input variables the grades for complexity, comprehension and both at the same time. The best performing models were the ones achieved from all the eye tracking measures with feature selection applied to them combined with both complexity and comprehension estimates, provided by the users. Notably, the highest accuracy was achieved when only comprehension was used for the classification. Modelling interest through textual complexity and comprehension significantly improves results across the board, judging by the higher achieved accuracy. Since the premise is held true, next we aim to model comprehension and complexity separately, and use predicted values for the same approach. This way, by applying supervised learning using explicit feedback, we can predict implicit feedback the same way solely from the eye tracking signal.

Table 4. The highest classification accuracy obtained for the two datasets with and without complexity and comprehension grades provided by participants. Bold cells represent highest values per variable set.

Dataset	Results from ETS	ETS + complexity	ETS + comprehension	ETS + complexity + comprehension
No ETS		81.96%	82.48%	79.97%
Processed	83.51%	**84.35%**	**86.47%**	85.83%
Normalized	**83.67%**	83.54%	86.07%	**85.85%**

6.3.2. Description of results in predicting complexity/comprehension using regression models

After we established the relevance of complexity and comprehension grades, we aim to model these values using the ETS and the features we extracted from it. In order to do that, we train and test two separate regression models: complexity model and comprehension model. User provided grades are on the scale of 1 to 7 and we use the same scale for the regression. We evaluate the performance of the regression models using Mean Squared Error (MSE). Also, we evaluate the relative accuracy, by counting portion of the test cases, where the predicted value is no more than 1 point away from the true value, e.g. predicting 2 or 3.5 for the true value of 3 would be considered a success, while 1.9 or 5 would be considered failures. Values of the MSE and relative accuracy are presented in Table 5 and 6. Despite the fact, that more than half of the values are predicted relatively accurately, we can suggest that regression of a variable of subjective nature with a fairly small samples size does not lead the perfect model. However, modelling complexity and comprehension is not the primary goal of this research, and we aim to use these predicted values for the final interest model and see whether they do lead to the improvement of the results.

Table 5. The smallest Mean Squared Error for regression models for complexity and comprehension for both datasets.

Dataset	Complexity	Comprehension	Interest
Processed	1.96	1.34	1.76
Normalized	1.92	1.38	1.63

Table 6. Relative accuracy for regression models for complexity and comprehension for both datasets. Whenever prediction error is less or equal to 1, the prediction is considered successful.

Dataset	Complexity	Comprehension	Interest
Processed	52.91%	66.76%	57.56%
Normalized	52.74%	67.58%	58.54%

6.3.3. Description of results using predicted complexity/comprehension

The final model is constructed by combining complexity and comprehension regression models, that predict respective values based on the eye tracking variables. Next, complexity and comprehension estimates are added to the variables that are fed to the interest classification models. It should be noted, that the data is split for cross validation for the full pipeline of the modelling process. Meaning that whenever complexity, comprehension or interest grade provided by the user is in the training set, so are all of the 'true' values, provided by the user. This is done to ensure consistency and simulate real-world scenarios, where none of the true values are known for the test cases.

Firstly, we test feeding the regression models the same set of variables that are used for classification model of interest. Since feature selection is implemented as a wrapper method of best-first search, adding nested feature selection for complexity/comprehension is not computationally feasible. In Table 7 we present the highest classification accuracy that was achieved using predicted values of complexity and/or comprehension obtain with this method. Overall, the only improvement in accuracy was observed when predicting comprehension for the texts in the normalized dataset. Due to inaccuracy in the regression models using identical variable subset, results did not improve as much, as when using the complexity/comprehension grades reported by participants.

Table 7. The highest classification accuracy obtained for the two datasets with and without complexity and comprehension grades predicted by the regression models. Regression and classification is done using the same ETS variables, found from feature selection search methods. Bold cells represent highest values per variable set.

Dataset	Results from ETS	ETS + complexity	ETS + comprehension	ETS + complexity + comprehension
Processed	83.51%	**83.66%**	83.49%	83.34%
Normalized	**83.67%**	82.34%	**84.34%**	**83.36%**

However, we can chose the regression models for complexity and comprehension to use a selected variable subset at all times. After evaluating performance of the regression models using

exact same feature filtering and feature selection, as for the classification model, we find models that lead to the smallest MSE in the regression. It is also notable, that although RF overwhelmingly performed better than SVM for classification, both complexity and comprehension best performing models turned out to be SVM. Results of the models, using this implementation of regression models, can be found in Table 8. Overall, the results improve from adding both complexity and comprehension estimates, however using both prediction at once does not yield accuracy improvement. Nevertheless, the highest classification accuracy for predicting perceived interest using solely Eye Tracking Measurements is achieved by constructing a regression model for comprehension and using predicted value in combination with ETS features for the final classification model.

Table 8. The highest classification accuracy obtained for the two datasets with and without complexity and comprehension grades predicted by the regression models. Prediction is done using best performing set of variables, classification variable selection is done through feature selection search methods. Bold cells represent highest values per variable set.

Dataset	Results from ETS	ETS + complexity	ETS + comprehension	ETS + complexity + comprehension
Processed	83.51%	**83.94%**	**84.35%**	82.94%
Normalized	**83.67%**	83.04%	84.09%	**83.61%**

7. General discussion

Results of the classification process can be considered somewhat successful, though there are a lot of caveats to that. Firstly, the nature of uncertainty what can be considered interesting or not interesting, influences the balance of the data. The best accuracy was achieved using the cut-off between interesting and not interesting in the middle of the grading scale. Meanwhile splitting the classes in a more balanced way only reduced the performance of the models. The fact that applying commonly used resampling method ROSE reduced the performance of the model even further, we question whether the set of measurements acquired from ETS can accomodate for precise classification between interesting and not interesting texts that are read. In the dataset where 81.88% text readings are classified as interesting, the models that yield the best accuracy predict roughly 90% of the texts to be interesting (for exact values refer to Appendix B). We think that given a larger sample size and a clearer indication of the feedback by the participants, the proposed models can be improved and capture the implicit feedback of perceived interest. Also, the quality of data is very influential on the analysis that it can be used for. Not all the features of the eye tracking signal could be studied in our case, and the accuracy of the data could influence all the measures that were calculated and used for the modelling process.

Importantly, the results improved when we incorporated the approach proposed by van der Sluis et al in [1] of modelling interest through complexity and comprehension. Although predicting the values of complexity and comprehension was not precise, the predicted values did improve the performance of the interest classification model. We believe that this approach can be further studied using eye tracking data or combining eye tracking with other approaches, such as text mining. Our project shows that eye tracking data can be effectively used on its own for modelling perceived interest and this may help overcome the challenge of affective computing in overcoming the barrier between emotions and computer recognition of them. Although, eye tracking data was previously used for estimating complexity (e.g.[32]) or cognitive load in general, our study proposes using modelling aggregated features from eye tracking data, that in our case were complexity and comprehension, and later use these values for another model.

Despite not achieving stellar accuracy rates, we believe that the study is still relevant and useful for further applications. For instance, the framework with GUI that was developed, can be used for another set of the eye tracking data, or even adapted to work on-the-fly with a device to record the signal. The original intent for the project was to use a mobile eye tracker and compare recorded results to the dataset, that was provided earlier. However, it ended up being out of the scope for the project and it remains to be seen what can be achieved when the data is acquired in the controlled environment where it can be studied and evaluated straightaway with our framework.

However, even for the data that was used, we believe that our structured approach to use feature engineering can prove to be useful. In particular, in our tests, such variables as fixation duration kurtosis and saccade duration kurtosis ended up being widely selected by the feature selection methods despite not being recommended in any related literature we found. In our opinion, even the variable set that is produced for this particular dataset can be used for building models with higher accuracy with better knowledge and implementation of machine learning methods. Due to the fact, that building of the models was just a part of the project, rather than its focus, we did not realize the potential that our approach can facilitate for. As a takeaway for future studies we may provide is that sometimes using less obvious and well-documented features of the studied data can turn out to be beneficial if no measurements are ignored. Much like Guyon and Elisseeff mention in [25], sometimes combination of correlated or seemingly redundant variables may lead to an improvement in results.

8. Conclusion

In this study, we explored application of eye tracking signal to the approach of affective computing in the problem of predicting user's implicit feedback in a form of perceived interest. We provided a structured approach towards processing the signal and extracting valuable features from it. Next, these features were quantified using statistical measurements and used for the classification model, that would predict whether an article was interesting or not. We reached mean classification accuracy of 84.35% by modelling interest through first evaluating comprehension of the text. By adopting the approach from [1] and applying it to the eye tracking data, we propose potential applications for using eye movements for affective signal processing. We developed a framework with an interface in order to process and analyze the data. Where we implemented methods that are described in this thesis.

9. References

[1] Van der Sluis, F., van den Broek, E.L., Glassey, R.J., Dijk, E.M. and Jong, F.M., 2014. W hen complexity becomes interesting. Journal of the Association for Information Science and Technology, 65(7), pp.1478-1500.

[2] Van den Broek, E.L., 2011. Affective Signal Processing (ASP): Unraveling the mystery of emotions. PhD Thesis, Centre for Telematics and Information Technology University of Twente, Enschede

[3] Holmqvist, K., Nyström, M., Andersson, R., Dewhurst, R., Jarodzka, H. and Van de Weijer, J., 2011. *Eye tracking: A comprehensive guide to methods and measures*. OUP Oxford.

[4] Holmqvist, K., & Andersson, R. 2017. Eye tracking: a comprehensive guide to methods, paradigms, and measures.

[5] Stampe, D.M., 1993. Heuristic filtering and reliable calibration methods for video-based pupil-tracking systems. *Behavior Research Methods, Instruments, & Computers*, *25*(2), pp.137-142.

[6] Salvucci, D.D. and Goldberg, J.H., 2000, November. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 symposium on Eye tracking research & applications* (pp. 71-78). ACM.

[7] Nyström, M. and Holmqvist, K., 2010. An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data. *Behavior research methods*, *42*(1), pp.188-204.

[8] Blascheck, T., Kurzhals, K., Raschke, M., Burch, M., Weiskopf, D. and Ertl, T., 2017, December. Visualization of eye tracking data: A taxonomy and survey. In *Computer Graphics Forum* (Vol. 36, No. 8, pp. 260-284).

[9] Holmqvist, K., Nyström, M. and Mulvey, F., 2012, March. Eye tracker data quality: what it is and how to measure it. In *Proceedings of the symposium on eye tracking research and applications* (pp. 45-52). ACM.

[10] Blignaut, P. and Wium, D., 2014. Eye-tracking data quality as affected by ethnicity and experimental design. *Behavior research methods*, *46*(1), pp.67-80.

[11] Ooms, K., Dupont, L., Lapon, L. and Popelka, S., 2015. Accuracy and precision of fixation locations recorded with the low-cost Eye Tribe tracker in different experimental setups. *Journal of eye movement research*, *8*(1).

[12] Niehorster, D.C., Cornelissen, T.H., Holmqvist, K., Hooge, I.T. and Hessels, R.S., 2018. What to expect from your remote eye-tracker when participants are unrestrained. *Behavior research methods*, *50*(1), pp.213-227.

[13] Gibaldi, A., Vanegas, M., Bex, P.J. and Maiello, G., 2017. Evaluation of the Tobii EyeX Eye tracking controller and Matlab toolkit for research. *Behavior research methods*, *49*(3), pp.923-946.

[14] Titz, J., Scholz, A. and Sedlmeier, P., 2017. Comparing eye trackers by correlating their eye-metric data. *Behavior research methods*, pp.1-11.

[15] Wang, D., Mulvey, F.B., Pelz, J.B. and Holmqvist, K., 2017. A study of artificial eyes for the measurement of precision in eye-trackers. *Behavior research methods*, *49*(3), pp.947-959.

[16] Olsen, A., 2012. The Tobii I-VT fixation filter. *Tobii Technology*.

[17] Leppänen, J.M., Forssman, L., Kaatiala, J., Yrttiaho, S. and Wass, S., 2015. Widely applicable MATLAB routines for automated analysis of saccadic reaction times. *Behavior Research Methods*, *47*(2), pp.538-548.

[18] Wass, S.V., Smith, T.J. and Johnson, M.H., 2013. Parsing eye-tracking data of variable quality to provide accurate fixation duration estimates in infants and adults. *Behavior Research Methods*, *45*(1), pp.229-250.

[19] Komogortsev, O., Gobert, D.V. and Dai, Z., 2010. Classification algorithm for saccadic oculomotor behavior.

[20] Duchowski, A.T., 2007. Eye tracking methodology. *Theory and practice*, *328*.

[21] Savitzky, A. and Golay, M.J., 1964. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, *36*(8), pp.1627-1639.

[22] Frank, M.C., Vul, E. and Johnson, S.P., 2009. Development of infants' attention to faces during the first year. *Cognition*, *110*(2), pp.160-170.

[23] Andersson, R., Larsson, L., Holmqvist, K., Stridh, M. and Nyström, M., 2017. One algorithm to rule them all? An evaluation and discussion of ten eye movement event-detection algorithms. Behavior research methods, 49(2), pp.616-637.

[24] Andersson, R., Nyström, M. and Holmqvist, K., 2010. Sampling frequency and eye-tracking measures: how speed affects durations, latencies, and more. Journal of Eye Movement Research, 3(3).

[25] Guyon, I. and Elisseeff, A., 2003. An introduction to variable and feature selection. Journal of machine learning research, 3(Mar), pp.1157-1182.

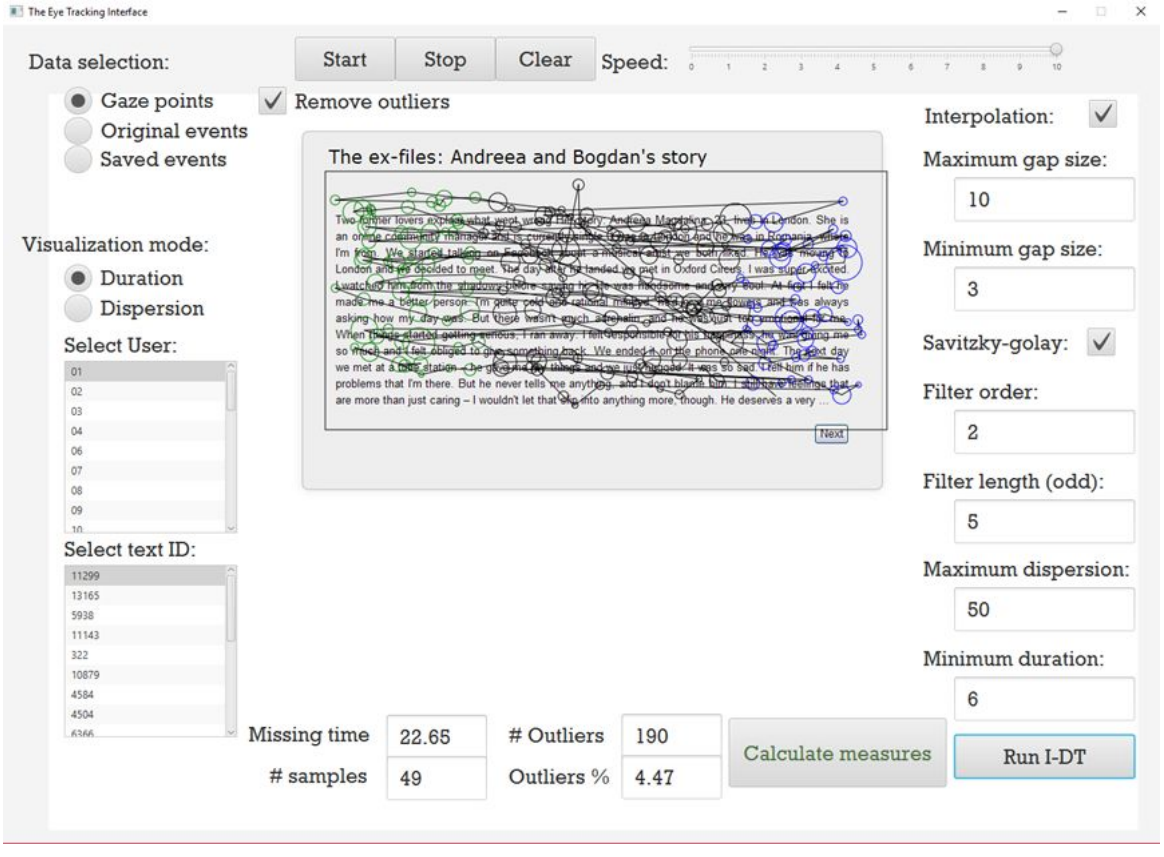
[26] Zemblys, R., 2016. Eye-movement event detection meets machine learning. BIOMEDICAL ENGINEERING 2016, 20(1).
[27] Zemblys, R., Niehorster, D.C., Komogortsev, O. and Holmqvist, K., 2018. Using machine learning to detect events in eye-tracking data. Behavior research methods, 50(1), pp.160-181
[28] Blignaut, P. (2009). Fixation identification: The optimum threshold for a dispersion algorithm. Attention, Perception, & Psychophysics, 71(4), 881–895.
[29] Di Russo, F., Pitzalis, S. and Spinelli, D., 2003. Fixation stability and saccadic latency in elite shooters. *Vision Research*, *43*(17), pp.1837-1845.
[30] Falkmer, T., Dahlman, J., Dukic, T., Bjällmark, A. and Larsson, M., 2008. Fixation identification in centroid versus start-point modes using eye-tracking data. Perceptual and motor skills, 106(3), pp.710-724.
[31] Den Buurman, R., Roersema, T. and Gerrissen, J.F., 1981. Eye movements and the perceptual span in reading. Reading Research Quarterly, pp.227-235.
[32] Medero, J., 2014. *Automatic Characterization of Text Difficulty*(Doctoral dissertation).
[33] Hyrskykari, A., 2006. Utilizing eye movements: Overcoming inaccuracy while tracking the focus of attention during reading. *Computers in human behavior*, *22*(4), pp.657-671.
[34] Munn, S.M., Stefano, L. and Pelz, J.B., 2008, August. Fixation-identification in dynamic scenes: Comparing an automated algorithm to manual coding. In *Proceedings of the 5th symposium on Applied perception in graphics and visualization* (pp. 33-42). ACM.
[35] Salojärvi, J., Puolamäki, K., Simola, J., Kovanen, L., Kojo, I. and Kaski, S., 2005. Inferring relevance from eye movements: Feature extraction.
[36] Holland, C. and Komogortsev, O.V., 2011, October. Biometric identification via eye movement scanpaths in reading. In *Biometrics (IJCB), 2011 International Joint Conference on* (pp. 1-8). IEEE.
[37] Chandrashekar, G. and Sahin, F., 2014. A survey on feature selection methods. *Computers & Electrical Engineering*, *40*(1), pp.16-28.
[38] Tabachnick, B.G. and Fidell, L.S., 2007. *Using multivariate statistics*. Allyn & Bacon/Pearson Education.
[39] Blascheck, T., Kurzhals, K., Raschke, M., Burch, M., Weiskopf, D. and Ertl, T., 2014, June. State-of-the-art of visualization for eye tracking data. In *Proceedings of EuroVis*(Vol. 2014).
[40] Holland, C.D. and Komogortsev, O.V., 2013, June. Complex eye movement pattern biometrics: Analyzing fixations and saccades. In Biometrics (ICB), 2013 International Conference on (pp. 1-8). IEEE
[41] Warga, M., Lüdtke, H., Wilhelm, H. and Wilhelm, B., 2009. How do spontaneous pupillary oscillations in light relate to light intensity?. Vision research, 49(3), pp.295-300.
[42] Schotter, E. R., & Rayner, K. (2013). Eye movement in reading Implications for reading subtitles. Eye tracking in audiovisual translation, 83-104.
[43] Neumann, U., Genze, N. and Heider, D., 2017. EFS: an ensemble feature selection tool implemented as R-package and web-application. *BioData mining*, *10*(1), p.21.
[44] Sun, Y., Wong, A.K. and Kamel, M.S., 2009. Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence*, *23*(04), pp.687-719.
[45] Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H. and Bing, G., 2017. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, *73*, pp.220-239.
[46] Fernández, A., Garcia, S., Herrera, F. and Chawla, N.V., 2018. SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary. *Journal of Artificial Intelligence Research*, *61*, pp.863-905.
[47] Lunardon, N., Menardi, G. and Torelli, N., 2014. ROSE: A Package for Binary Imbalanced Learning. *R Journal*, *6*(1).
[48] Neumann, U., Genze, N. and Heider, D., 2017. EFS: an ensemble feature selection tool implemented as R-package and web-application. *BioData mining*, *10*(1), p.21.

10. Appendix A. GUI implementation

Over the course of this project, we developed and improved a framework with Graphical User Interface (GUI), that lets users interact with the data and manipulate it using all of the implemented processing and visualization methods. Firstly, the program was designed to visualize the data, reported by the eye tracker, in order to evaluate the data in the context of the visual stimulus (the text image). However, once the processing pipeline started being implemented, the ability to directly compare the influence of a certain processing step on the data, helped with instant feedback on all of the methods that were implemented. Throughout the course of the project, using one big framework for all the data processing, helped with comparison and evaluation at every step.

Moreover, the data could be studied at the feature parameter level once feature engineering was implemented. This let us get even more insight into the influence of individual signal processing blocks that were executed. Getting results in line with expectations and literature would mean that everything is working properly, but sometimes getting unexpected results could unveil new features of the eye tracking signal. For example, visualizing outliers defined to be outside of the bounding box and adding up the total portion of the samples being outliers, help in evaluating the data quality of specific participants and the dataset overall. Also, when implementing line-to-line saccades, as proposed by Medero in [32], GUI helped by using different colors to label fixations that are used in the moving-window detection approach.

Figure 5. Screenshot of the GUI with the framework, where user can choose: data to be read; replay speed; outlier removal; interpolation and its parameters; Savitzky-Golay filter and its parameters; i-DT and its parameters.



Implementation of the framework was done in Java with the use of JavaFX as the primary GUI, and integration with R using rJava and JRI (Java-R Interface) to feed the data directly from the program to R and back for processing. Source code for the implementation can be found at GitHub as a 'Mockup' together with other tools that were used throughout the project.

11. Appendix B. Test results

$$TP = True\ Positives,\ FP = False\ Positives$$
$$TN = True\ Negatives,\ FN = False\ Negatives$$

$$\%\ of\ P = Portion\ of\ Positives\ in\ predictions = \frac{TP + FP}{TP + FP + TN + FN}$$
$$\%\ of\ N = Portion\ of\ Negatives\ in\ predictions = \frac{TN + FN}{TP + FP + TN + FN}$$
$$TPR = True\ Positive\ Rate = \frac{TP}{TP + FP}$$
$$FNR = True\ Negative\ Rate = \frac{TN}{TN + FN}$$

$$t_i = Threshold\ of\ Interest\ (see\ 5.3.1)$$

No eye tracking signal

1 - used, 0 - not used

t_i	Model	Complexity	Comprehension	Accuracy	% of P	TPR	% of N	TNR
3.5	RF	1	0	80.68	95.53	82.78	4.47	24.25
3.5	SVM	1	0	81.96	98.51	82.4	1.49	18
3.5	RF	0	1	81.69	93.13	84.23	6.87	42.1
3.5	SVM	0	1	82.48	95.69	83.67	4.31	40.92
3.5	RF	1	1	79.97	87.81	85.21	12.19	44.26
3.5	SVM	1	1	78.83	86.88	85	13.12	38.68
4	RF	1	0	69.09	92.39	70.4	7.61	49.92
4	SVM	1	0	66.08	91.08	69.13	8.92	27.8
4	RF	0	1	70.12	69.11	78.62	30.89	53.35
4	SVM	0	1	69.96	70.36	78.65	29.64	53.81
4	RF	1	1	72.07	71.16	78.54	28.24	56.38
4	SVM	1	1	73.67	74.65	78.4	25.35	59.92
4.5	RF	1	0	61.92	71.54	65.54	28.46	54.87
4.5	SVM	1	0	57.8	73.24	62.6	26.76	49.77
4.5	RF	0	1	73.87	59.69	78.46	40.31	68.12
4.5	SVM	0	1	73.07	61.03	77.12	38.97	67.01
4.5	RF	1	1	72.46	63.43	75.55	36.57	67.73

4.5	SVM	1	1	71.15	64	74.35	36	66.5

Processed dataset

Dataset	t_i	Feature filter	Feature selection	Model	Balancing	Scoring function	Accuracy	% of P	TPR	% of N	TNR
processed	3.5	no	best-first	RF	no	accuracy	83.51	97.22	83.62	2.78	47.17
processed	3.5	no	forward	RF	no	accuracy	83.51	97.22	83.62	2.78	47.17
processed	3.5	no	best-first	RF	no	zeroes	81.55	91.75	84.58	8.25	51.06
processed	3.5	no	forward	RF	no	zeroes	81.55	91.75	84.58	8.25	51.06
processed	3.5	0.3	best-first	RF	no	accuracy	83.8	95.24	84.49	4.76	61.08
processed	3.5	0.3	forward	RF	no	accuracy	83.8	95.24	84.49	4.76	61.08
processed	3.5	0.3	best-first	RF	no	zeroes	81.55	91.75	84.58	8.25	51.06
processed	3.5	0.3	forward	RF	no	zeroes	81.55	91.75	84.58	8.25	51.06
processed	3.5	no	best-first	RF	ROSE	accuracy	74.84	81.57	84.88	18.43	29.84
processed	3.5	no	forward	RF	ROSE	accuracy	74.84	81.57	84.88	18.43	29.84
processed	3.5	no	best-first	RF	ROSE	zeroes	60.22	55.51	87.99	44.49	26.16
processed	3.5	no	forward	RF	ROSE	zeroes	60.22	55.51	87.99	44.49	26.16
processed	3.5	0.3	best-first	RF	ROSE	accuracy	74.84	81.57	84.88	18.43	29.84
processed	3.5	0.3	forward	RF	ROSE	accuracy	74.84	81.57	84.88	18.43	29.84
processed	3.5	0.3	best-first	RF	ROSE	zeroes	60.22	55.51	87.99	44.49	26.16
processed	3.5	0.3	forward	RF	ROSE	zeroes	60.22	55.51	87.99	44.49	26.16
processed	3.5	no	best-first	SVM	no	accuracy	82.63	97.12	83.19	2.88	37.95
processed	3.5	no	forward	SVM	no	accuracy	82.63	97.12	83.19	2.88	37.95
processed	3.5	no	best-first	SVM	no	zeroes	81.52	90.77	84.89	9.23	46.23
processed	3.5	no	forward	SVM	no	zeroes	81.52	90.77	84.89	9.23	46.23
processed	3.5	0.3	best-first	SVM	no	accuracy	82.63	97.12	83.19	2.88	37.95
processed	3.5	0.3	forward	SVM	no	accuracy	82.63	97.12	83.19	2.88	37.95
processed	3.5	0.3	best-first	SVM	no	zeroes	82.15	93.09	84.34	6.91	42.32
processed	3.5	0.3	forward	SVM	no	zeroes	82.15	93.09	84.34	6.91	42.32
processed	3.5	no	best-first	SVM	ROSE	accuracy	69.68	76.91	83.58	23.09	23.14
processed	3.5	no	forward	SVM	ROSE	accuracy	69.68	76.91	83.58	23.09	23.14
processed	3.5	no	best-first	SVM	ROSE	zeroes	64.13	65.89	84.93	34.11	28.05
processed	3.5	no	forward	SVM	ROSE	zeroes	64.13	65.89	84.93	34.11	28.05
processed	3.5	0.3	best-first	SVM	ROSE	accuracy	69.68	76.91	83.58	23.09	23.14
processed	3.5	0.3	forward	SVM	ROSE	accuracy	69.68	76.91	83.58	23.09	23.14
processed	3.5	0.3	best-first	SVM	ROSE	zeroes	64.13	65.89	84.93	34.11	28.05
processed	3.5	0.3	forward	SVM	ROSE	zeroes	64.13	65.89	84.93	34.11	28.05
processed	4	no	best-first	RF	no	accuracy	73.48	89.3	73.63	10.7	72.04
processed	4	no	forward	RF	no	accuracy	73.48	89.3	73.63	10.7	72.04
processed	4	no	best-first	RF	no	zeroes	71.81	88.16	73.02	11.84	67.06

processed	4	no	forward	RF	no	zeroes	71.81	88.16	73.02	11.84	67.06
processed	4	0.3	best-first	RF	no	accuracy	73.66	88.33	74.04	11.67	69.61
processed	4	0.3	forward	RF	no	accuracy	73.66	88.33	74.04	11.67	69.61
processed	4	0.3	best-first	RF	no	zeroes	71.81	88.16	73.02	11.84	67.06
processed	4	0.3	forward	RF	no	zeroes	71.81	88.16	73.02	11.84	67.06
processed	4	no	best-first	RF	ROSE	accuracy	65.39	83.46	67.76	16.54	37.11
processed	4	no	forward	RF	ROSE	accuracy	65.39	83.46	67.76	16.54	37.11
processed	4	no	best-first	RF	ROSE	zeroes	63.99	76.96	69.72	23.04	45.13
processed	4	no	forward	RF	ROSE	zeroes	63.99	76.96	69.72	23.04	45.13
processed	4	0.3	best-first	RF	ROSE	accuracy	65.07	76.57	71.93	23.43	42.44
processed	4	0.3	forward	RF	ROSE	accuracy	65.07	76.57	71.93	23.43	42.44
processed	4	0.3	best-first	RF	ROSE	zeroes	61.9	71.5	71.74	28.2	41.25
processed	4	0.3	forward	RF	ROSE	zeroes	61.9	71.5	71.74	28.2	41.25
processed	4	no	best-first	SVM	no	accuracy	71.66	82.67	74.37	17.33	59.6
processed	4	no	forward	SVM	no	accuracy	71.66	82.67	74.37	17.33	59.6
processed	4	no	best-first	SVM	no	zeroes	71.66	82.67	74.37	17.33	59.6
processed	4	no	forward	SVM	no	zeroes	71.66	82.67	74.37	17.33	59.6
processed	4	0.3	best-first	SVM	no	accuracy	71.66	82.67	74.37	17.33	59.6
processed	4	0.3	forward	SVM	no	accuracy	71.66	82.67	74.37	17.33	59.6
processed	4	0.3	best-first	SVM	no	zeroes	71.66	82.67	74.37	17.33	59.6
processed	4	0.3	forward	SVM	no	zeroes	71.66	82.67	74.37	17.33	59.6
processed	4	no	best-first	SVM	ROSE	accuracy	64.06	83.53	64.81	16.47	35.09
processed	4	no	forward	SVM	ROSE	accuracy	64.06	83.53	64.81	16.47	35.09
processed	4	no	best-first	SVM	ROSE	zeroes	59.93	63.17	72.79	36.83	42.91
processed	4	no	forward	SVM	ROSE	zeroes	59.93	63.17	72.79	36.83	42.91
processed	4	0.3	best-first	SVM	ROSE	accuracy	62.48	85.26	62.4	14.74	18.9
processed	4	0.3	forward	SVM	ROSE	accuracy	62.48	85.26	62.4	14.74	18.9
processed	4	0.3	best-first	SVM	ROSE	zeroes	59.93	63.17	72.79	36.83	42.91
processed	4	0.3	forward	SVM	ROSE	zeroes	59.93	63.17	72.79	36.83	42.91
processed	4.5	no	best-first	RF	no	accuracy	66.15	69.34	68.98	30.66	61.72
processed	4.5	no	forward	RF	no	accuracy	66.15	69.34	68.98	30.66	61.72
processed	4.5	no	best-first	RF	no	zeroes	62.29	74.47	67.58	25.53	64.85
processed	4.5	no	forward	RF	no	zeroes	62.29	74.47	67.58	25.53	64.85
processed	4.5	0.3	best-first	RF	no	accuracy	66.15	69.34	68.98	30.66	61.72
processed	4.5	0.3	forward	RF	no	accuracy	66.15	69.34	68.98	30.66	61.72
processed	4.5	0.3	best-first	RF	no	zeroes	65.78	75.58	66.87	24.42	62.97
processed	4.5	0.3	forward	RF	no	zeroes	65.78	75.58	66.87	24.42	62.97
processed	4.5	no	best-first	RF	ROSE	accuracy	61.98	96.42	61.34	3.58	46.72
processed	4.5	no	forward	RF	ROSE	accuracy	61.98	96.42	61.34	3.58	46.72
processed	4.5	no	best-first	RF	ROSE	zeroes	61.42	90.56	61.83	9.44	54
processed	4.5	no	forward	RF	ROSE	zeroes	61.42	90.56	61.83	9.44	54
processed	4.5	0.3	best-first	RF	ROSE	accuracy	62.68	93.89	61.67	6.11	61.67

processed	4.5	0.3	forward	RF	ROSE	accuracy	62.68	93.89	61.67	6.11	61.67
processed	4.5	0.3	best-first	RF	ROSE	zeroes	61.89	90.75	61.87	9.25	61.97
processed	4.5	0.3	forward	RF	ROSE	zeroes	61.89	90.75	61.87	9.25	61.97
processed	4.5	no	best-first	SVM	no	accuracy	65.11	80.23	65.51	19.77	65.17
processed	4.5	no	forward	SVM	no	accuracy	65.11	80.23	65.51	19.77	65.17
processed	4.5	no	best-first	SVM	no	zeroes	63.89	85.37	63.88	14.63	65.21
processed	4.5	no	forward	SVM	no	zeroes	63.89	85.37	63.88	14.63	65.21
processed	4.5	0.3	best-first	SVM	no	accuracy	64.26	82.92	64.66	17.08	64.86
processed	4.5	0.3	forward	SVM	no	accuracy	64.26	82.92	64.66	17.08	64.86
processed	4.5	0.3	best-first	SVM	no	zeroes	63.89	85.37	63.88	14.63	65.21
processed	4.5	0.3	forward	SVM	no	zeroes	63.89	85.37	63.88	14.63	65.21
processed	4.5	no	best-first	SVM	ROSE	accuracy	61.98	96.42	61.34	3.58	46.72
processed	4.5	no	forward	SVM	ROSE	accuracy	61.98	96.42	61.34	3.58	46.72
processed	4.5	no	best-first	SVM	ROSE	zeroes	61.42	90.56	61.83	9.44	54
processed	4.5	no	forward	SVM	ROSE	zeroes	61.42	90.56	61.83	9.44	54
processed	4.5	0.3	best-first	SVM	ROSE	accuracy	62.85	92.52	62.19	7.48	58.4
processed	4.5	0.3	forward	SVM	ROSE	accuracy	62.85	92.52	62.19	7.48	58.4
processed	4.5	0.3	best-first	SVM	ROSE	zeroes	60.57	85.86	62.03	14.14	54.24
processed	4.5	0.3	forward	SVM	ROSE	zeroes	60.57	85.86	62.03	14.14	54.24

Normalized dataset

Dataset	t_i	Feature filter	Feature selection	Model	Balancing	Scoring function	Accuracy	% of P	TPR	% of N	TNR
normalized	3.5	no	best-first	RF	no	accuracy	83.67	94.51	84.65	5.49	59.17
normalized	3.5	no	forward	RF	no	accuracy	83.67	94.51	84.65	5.49	59.17
normalized	3.5	no	best-first	RF	no	zeroes	80.97	91.95	84.25	8.05	41.75
normalized	3.5	no	forward	RF	no	zeroes	80.97	91.95	84.25	8.05	41.75
normalized	3.5	0.3	best-first	RF	no	accuracy	83.67	94.51	84.65	5.49	59.17
normalized	3.5	0.3	forward	RF	no	accuracy	83.67	94.51	84.65	5.49	59.17
normalized	3.5	0.3	best-first	RF	no	zeroes	83.67	94.51	84.65	5.49	59.17
normalized	3.5	0.3	forward	RF	no	zeroes	83.67	94.51	84.65	5.49	59.17
normalized	3.5	no	best-first	RF	ROSE	accuracy	70.26	70.08	87.3	29.92	31.7
normalized	3.5	no	forward	RF	ROSE	accuracy	70.26	70.08	87.3	29.92	31.7
normalized	3.5	no	best-first	RF	ROSE	zeroes	68.38	65.96	88.19	34.04	31.64
normalized	3.5	no	forward	RF	ROSE	zeroes	68.38	65.96	88.19	34.04	31.64
normalized	3.5	0.3	best-first	RF	ROSE	accuracy	68.24	67.26	87.16	32.74	29.31
normalized	3.5	0.3	forward	RF	ROSE	accuracy	68.24	67.26	87.16	32.74	29.31
normalized	3.5	0.3	best-first	RF	ROSE	zeroes	69.49	67.57	87.98	32.43	31.25
normalized	3.5	0.3	forward	RF	ROSE	zeroes	69.49	67.57	87.98	32.43	31.25
normalized	3.5	no	best-first	SVM	no	accuracy	81.99	97.89	82.63	2.11	27.33
normalized	3.5	no	forward	SVM	no	accuracy	81.99	97.89	82.63	2.11	27.33

normalized	3.5	no	best-first	SVM	no	zeroes	79.66	92.54	83.21	7.46	35.56
normalized	3.5	no	forward	SVM	no	zeroes	79.66	92.54	83.21	7.46	35.56
normalized	3.5	0.3	best-first	SVM	no	accuracy	81.99	97.89	82.63	2.11	27.33
normalized	3.5	0.3	forward	SVM	no	accuracy	81.99	97.89	82.63	2.11	27.33
normalized	3.5	0.3	best-first	SVM	no	zeroes	78.77	90.63	83.47	9.37	34.38
normalized	3.5	0.3	forward	SVM	no	zeroes	78.77	90.63	83.47	9.37	34.38
normalized	3.5	no	best-first	SVM	ROSE	accuracy	64.3	64.01	86.31	35.99	26.2
normalized	3.5	no	forward	SVM	ROSE	accuracy	64.3	64.01	86.31	35.99	26.2
normalized	3.5	no	best-first	SVM	ROSE	zeroes	64.12	62.21	87.1	37.79	26.95
normalized	3.5	no	forward	SVM	ROSE	zeroes	64.12	62.21	87.1	37.79	26.95
normalized	3.5	0.3	best-first	SVM	ROSE	accuracy	64.12	62.21	87.1	37.79	26.95
normalized	3.5	0.3	forward	SVM	ROSE	accuracy	64.12	62.21	87.1	37.79	26.95
normalized	3.5	0.3	best-first	SVM	ROSE	zeroes	64.12	62.21	87.1	37.79	26.95
normalized	3.5	0.3	forward	SVM	ROSE	zeroes	64.12	62.21	87.1	37.79	26.95
normalized	4	no	best-first	RF	no	accuracy	69.86	79.16	74.43	20.84	52.61
normalized	4	no	forward	RF	no	accuracy	69.86	79.16	74.43	20.84	52.61
normalized	4	no	best-first	RF	no	zeroes	69.86	79.16	74.43	20.84	52.61
normalized	4	no	forward	RF	no	zeroes	69.86	79.16	74.43	20.84	52.61
normalized	4	0.3	best-first	RF	no	accuracy	70.78	84.73	73.47	15.27	57.06
normalized	4	0.3	forward	RF	no	accuracy	70.78	84.73	73.47	15.27	57.06
normalized	4	0.3	best-first	RF	no	zeroes	67.47	79.02	73.12	20.98	49.71
normalized	4	0.3	forward	RF	no	zeroes	67.47	79.02	73.12	20.98	49.71
normalized	4	no	best-first	RF	ROSE	accuracy	63.83	65.31	75.37	34.69	43.44
normalized	4	no	forward	RF	ROSE	accuracy	63.83	65.31	75.37	34.69	43.44
normalized	4	no	best-first	RF	ROSE	zeroes	64.22	60.28	77.63	39.72	44.59
normalized	4	no	forward	RF	ROSE	zeroes	64.22	60.28	77.63	39.72	44.59
normalized	4	0.3	best-first	RF	ROSE	accuracy	63.99	61.42	76.57	38.58	43.8
normalized	4	0.3	forward	RF	ROSE	accuracy	63.99	61.42	76.57	38.58	43.8
normalized	4	0.3	best-first	RF	ROSE	zeroes	62.75	59.04	76.82	40.96	43.91
normalized	4	0.3	forward	RF	ROSE	zeroes	62.75	59.04	76.82	40.96	43.91
normalized	4	no	best-first	SVM	no	accuracy	70.51	95.09	70.65	4.91	51.98
normalized	4	no	forward	SVM	no	accuracy	70.51	95.09	70.65	4.91	51.98
normalized	4	no	best-first	SVM	no	zeroes	70.1	91.64	71.23	8.36	57.25
normalized	4	no	forward	SVM	no	zeroes	70.1	91.64	71.23	8.36	57.25
normalized	4	0.3	best-first	SVM	no	accuracy	69.22	97.64	69.49	2.36	33.14
normalized	4	0.3	forward	SVM	no	accuracy	69.22	97.64	69.49	2.36	33.14
normalized	4	0.3	best-first	SVM	no	zeroes	70.19	79.26	74.62	20.74	55.28
normalized	4	0.3	forward	SVM	no	zeroes	70.19	79.26	74.62	20.74	55.28
normalized	4	no	best-first	SVM	ROSE	accuracy	59.93	62.55	72.84	37.45	41.47
normalized	4	no	forward	SVM	ROSE	accuracy	59.93	62.55	72.84	37.45	41.47
normalized	4	no	best-first	SVM	ROSE	zeroes	59.93	62.55	72.84	37.45	41.47
normalized	4	no	forward	SVM	ROSE	zeroes	59.93	62.55	72.84	37.45	41.47

normalized	4	0.3	best-first	SVM	ROSE	accuracy	59.93	62.55	72.84	37.45	41.47
normalized	4	0.3	forward	SVM	ROSE	accuracy	59.93	62.55	72.84	37.45	41.47
normalized	4	0.3	best-first	SVM	ROSE	zeroes	59.93	62.55	72.84	37.45	41.47
normalized	4	0.3	forward	SVM	ROSE	zeroes	59.93	62.55	72.84	37.45	41.47
normalized	4.5	no	best-first	RF	no	accuracy	66.15	69.34	68.98	30.66	61.72
normalized	4.5	no	forward	RF	no	accuracy	66.15	69.34	68.98	30.66	61.72
normalized	4.5	no	best-first	RF	no	zeroes	66.29	74.47	67.58	25.53	64.85
normalized	4.5	no	forward	RF	no	zeroes	66.29	74.47	67.58	25.53	64.85
normalized	4.5	0.3	best-first	RF	no	accuracy	66.15	69.34	68.98	30.66	61.72
normalized	4.5	0.3	forward	RF	no	accuracy	66.15	69.34	68.98	30.66	61.72
normalized	4.5	0.3	best-first	RF	no	zeroes	65.78	75.58	66.87	24.42	62.94
normalized	4.5	0.3	forward	RF	no	zeroes	65.78	75.58	66.87	24.42	62.94
normalized	4.5	no	best-first	RF	ROSE	accuracy	61.98	96.42	61.34	3.58	46.72
normalized	4.5	no	forward	RF	ROSE	accuracy	61.98	96.42	61.34	3.58	46.72
normalized	4.5	no	best-first	RF	ROSE	zeroes	61.42	90.56	61.83	9.44	54
normalized	4.5	no	forward	RF	ROSE	zeroes	61.42	90.56	61.83	9.44	54
normalized	4.5	0.3	best-first	RF	ROSE	accuracy	62.68	93.89	61.87	6.11	61.27
normalized	4.5	0.3	forward	RF	ROSE	accuracy	62.68	93.89	61.87	6.11	61.27
normalized	4.5	0.3	best-first	RF	ROSE	zeroes	61.89	90.75	61.87	9.25	61.97
normalized	4.5	0.3	forward	RF	ROSE	zeroes	61.89	90.75	61.87	9.25	61.97
normalized	4.5	no	best-first	SVM	no	accuracy	65.11	80.23	65.51	19.77	65.17
normalized	4.5	no	forward	SVM	no	accuracy	65.11	80.23	65.51	19.77	65.17
normalized	4.5	no	best-first	SVM	no	zeroes	63.89	85.37	63.88	14.63	65.21
normalized	4.5	no	forward	SVM	no	zeroes	63.89	85.37	63.88	14.63	65.21
normalized	4.5	0.3	best-first	SVM	no	accuracy	64.26	82.92	64.66	17.08	64.86
normalized	4.5	0.3	forward	SVM	no	accuracy	64.26	82.92	64.66	17.08	64.86
normalized	4.5	0.3	best-first	SVM	no	zeroes	63.89	85.37	63.88	14.63	65.21
normalized	4.5	0.3	forward	SVM	no	zeroes	63.89	85.37	63.88	14.63	65.21
normalized	4.5	no	best-first	SVM	ROSE	accuracy	62.09	84.92	63	15.08	59.3
normalized	4.5	no	forward	SVM	ROSE	accuracy	62.09	84.92	63	15.08	59.3
normalized	4.5	no	best-first	SVM	ROSE	zeroes	60.82	89.06	60.93	10.94	57.78
normalized	4.5	no	forward	SVM	ROSE	zeroes	60.82	89.06	60.93	10.94	57.78
normalized	4.5	0.3	best-first	SVM	ROSE	accuracy	62.85	92.52	62.19	7.48	58.4
normalized	4.5	0.3	forward	SVM	ROSE	accuracy	62.85	92.52	62.19	7.48	58.4
normalized	4.5	0.3	best-first	SVM	ROSE	zeroes	60.57	85.86	62.03	14.14	54.24
normalized	4.5	0.3	forward	SVM	ROSE	zeroes	60.57	85.86	62.03	14.14	54.24

Processed dataset combined with complexity and comprehension

1 - original values (reported by participants)
2 - predicted values with regression models using the same feature set as interest

3 - predited values with regression models using optimized feature set
Green cells represent values for which accuracy is higher than highest value obtained with ETS only (83.8% for the case at line 2).

Dataset	t_i	Feature selection	Scoring function	Feature filter	Model	Complexity	Comprehension	Accuracy	% of P	TPR	% of N	TNR
processed	3.5	best-first	accuracy	0	RF	0	0	83.51	97.22	83.62	2.78	47.17
processed	3.5	best-first	accuracy	0.3	RF	0	0	83.8	95.24	84.49	4.76	61.08
processed	3.5	best-first	accuracy	0	SVM	0	0	82.63	97.12	83.19	2.88	37.95
processed	3.5	best-first	accuracy	0.3	SVM	0	0	82.63	97.12	83.19	2.88	37.95
processed	3.5	best-first	accuracy	0	RF	1	0	84.35	93.13	85.54	6.87	65.27
processed	3.5	best-first	accuracy	0.3	RF	1	0	84.35	93.13	85.54	6.87	65.27
processed	3.5	best-first	accuracy	0	SVM	1	0	82.25	96.08	83.51	3.92	40.67
processed	3.5	best-first	accuracy	0.3	SVM	1	0	82.25	96.08	83.51	3.92	40.67
processed	3.5	best-first	accuracy	0	RF	2	0	83.66	96.26	84.02	3.74	52.67
processed	3.5	best-first	accuracy	0.3	RF	2	0	83.66	96.26	84.02	3.74	52.67
processed	3.5	best-first	accuracy	0	SVM	2	0	82.18	97.08	83	2.92	33.42
processed	3.5	best-first	accuracy	0.3	SVM	2	0	82.18	97.08	83	2.92	33.42
processed	3.5	best-first	accuracy	0	RF	3	0	83.94	96.9	83.97	3.1	53.08
processed	3.5	best-first	accuracy	0.3	RF	3	0	82.24	96.98	83.07	3.02	33.83
processed	3.5	best-first	accuracy	0	SVM	3	0	81.95	98.21	82.5	1.79	22.5
processed	3.5	best-first	accuracy	0.3	SVM	3	0	81.63	98.89	82.13	1.11	9.33
processed	3.5	best-first	accuracy	0	RF	0	1	86.39	91.55	87.33	8.45	75.5
processed	3.5	best-first	accuracy	0.3	RF	0	1	86.47	91.82	87.23	8.18	75.85
processed	3.5	best-first	accuracy	0	SVM	0	1	85.01	89.97	87.19	10.03	61.48
processed	3.5	best-first	accuracy	0.3	SVM	0	1	85.01	89.97	87.19	10.03	61.48
processed	3.5	best-first	accuracy	0	RF	1	1	85.83	92.17	86.69	7.83	72.98
processed	3.5	best-first	accuracy	0.3	RF	1	1	85.83	92.17	86.69	7.83	72.98
processed	3.5	best-first	accuracy	0	SVM	1	1	81.75	85.09	87.36	14.91	49.4
processed	3.5	best-first	accuracy	0.3	SVM	1	1	79.87	83.58	86.89	16.42	41.54
processed	3.5	best-first	accuracy	0	RF	0	2	83.49	96.15	84	3.85	52.42
processed	3.5	best-first	accuracy	0.3	RF	0	2	83.49	96.15	84	3.85	52.42
processed	3.5	best-first	accuracy	0	SVM	0	2	81.69	99.86	81.8	0.14	0
processed	3.5	best-first	accuracy	0.3	SVM	0	2	81.28	95.02	83.31	4.98	36.1
processed	3.5	best-first	accuracy	0	RF	2	2	83.34	97.45	83.44	2.55	41.33
processed	3.5	best-first	accuracy	0.3	RF	2	2	83.06	97.87	83.16	2.13	35
processed	3.5	best-first	accuracy	0	SVM	2	2	79	92.24	83.05	7.76	30.61
processed	3.5	best-first	accuracy	0.3	SVM	2	2	78.73	90.68	83.4	9.32	32.54
processed	3.5	best-first	accuracy	0	RF	0	3	84.35	93.13	85.54	6.87	65.27
processed	3.5	best-first	accuracy	0.3	RF	0	3	84.35	93.13	85.54	6.87	65.27
processed	3.5	best-first	accuracy	0	SVM	0	3	82.52	96.08	83.51	3.92	40.67
processed	3.5	best-first	accuracy	0.3	SVM	0	3	82.52	96.08	83.51	3.92	40.67
processed	3.5	best-first	accuracy	0	RF	3	3	82.94	98.51	82.91	1.49	32

processed	3.5	best-first	accuracy	0.3	RF	3	3	82.45	99.04	82.48	0.96	21.5
processed	3.5	best-first	accuracy	0	SVM	3	3	79.18	96.23	81.78	3.77	9.08
processed	3.5	best-first	accuracy	0.3	SVM	3	3	77.73	94.5	81.56	5.5	9.62

Normalized dataset combined with complexity and comprehension

1 - original values (reported by participants)
2 - predicted values with regression models using the same feature set as interest
3 - predited values with regression models using optimized feature set
Green cells represent values for which accuracy is higher than highest value obtained with ETS only (83.67% for the case at line 1 or 2).

Dataset	t_i	Feature selection	Scoring function	Feature filter	Model	Complexity	Comprehension	Accuracy	% of P	TPR	% of N	TNR
normalized	3.5	best-first	accuracy	0	RF	0	0	83.67	94.51	84.65	5.49	59.17
normalized	3.5	best-first	accuracy	0.3	RF	0	0	83.67	94.51	84.65	5.49	59.17
normalized	3.5	best-first	accuracy	0	SVM	0	0	81.99	97.89	82.63	2.11	27.33
normalized	3.5	best-first	accuracy	0.3	SVM	0	0	81.99	97.89	82.63	2.11	27.33
normalized	3.5	best-first	accuracy	0	RF	1	0	83.54	97.75	83.48	2.25	44.5
normalized	3.5	best-first	accuracy	0.3	RF	1	0	81.97	93.11	84.3	6.89	43.91
normalized	3.5	best-first	accuracy	0	SVM	1	0	80.01	90.8	84.03	9.2	36.65
normalized	3.5	best-first	accuracy	0.3	SVM	1	0	79.53	92.45	83.22	7.55	32.97
normalized	3.5	best-first	accuracy	0	RF	2	0	81.74	99.29	82.06	0.71	6.33
normalized	3.5	best-first	accuracy	0.3	RF	2	0	82.34	97.14	83.09	2.86	33.93
normalized	3.5	best-first	accuracy	0	SVM	2	0	81.17	97.76	82.24	2.24	15.17
normalized	3.5	best-first	accuracy	0.3	SVM	2	0	80.81	97.4	82.18	2.6	16.31
normalized	3.5	best-first	accuracy	0	RF	3	0	83.04	98.69	82.91	1.31	32
normalized	3.5	best-first	accuracy	0.3	RF	3	0	82.05	98.87	82.31	1.13	18.5
normalized	3.5	best-first	accuracy	0	SVM	3	0	81.34	98.61	82.08	1.39	11
normalized	3.5	best-first	accuracy	0.3	SVM	3	0	81.34	98.61	82.08	1.39	11
normalized	3.5	best-first	accuracy	0	RF	0	1	86.07	93.48	86.33	6.52	73.3
normalized	3.5	best-first	accuracy	0.3	RF	0	1	84.98	92.48	86.18	7.52	68.28
normalized	3.5	best-first	accuracy	0	SVM	0	1	83.76	90.58	86.24	9.42	58.15
normalized	3.5	best-first	accuracy	0.3	SVM	0	1	82.74	89.4	86.16	10.6	51.71
normalized	3.5	best-first	accuracy	0	RF	1	1	85.85	94.2	85.96	5.8	72.67
normalized	3.5	best-first	accuracy	0.3	RF	1	1	85.06	94.12	85.56	5.88	69.67
normalized	3.5	best-first	accuracy	0	SVM	1	1	81.22	98.43	82.05	1.57	9.5
normalized	3.5	best-first	accuracy	0.3	SVM	1	1	78.6	83.16	86.42	16.84	40.7
normalized	3.5	best-first	accuracy	0	RF	0	2	83.26	97.43	83.44	2.57	43.33
normalized	3.5	best-first	accuracy	0.3	RF	0	2	84.34	95.81	83.57	4.19	40.83
normalized	3.5	best-first	accuracy	0	SVM	0	2	81.08	95.91	82.78	4.09	28.73
normalized	3.5	best-first	accuracy	0.3	SVM	0	2	80.65	94.23	83.21	5.77	31.57
normalized	3.5	best-first	accuracy	0	RF	2	2	83.36	97.56	83.47	2.44	41.33

normalized	3.5	best-first	accuracy	0.3	RF	2	2	82.36	96.87	83.18	3.13	36.08
normalized	3.5	best-first	accuracy	0	SVM	2	2	82.58	98.75	82.63	1.25	21.83
normalized	3.5	best-first	accuracy	0.3	SVM	2	2	82.02	99.18	82.19	0.82	13
normalized	3.5	best-first	accuracy	0	RF	0	3	84.09	97.26	83.92	2.74	51.33
normalized	3.5	best-first	accuracy	0.3	RF	0	3	82.06	97.62	82.78	2.38	25
normalized	3.5	best-first	accuracy	0	SVM	0	3	81.7	95.69	83.22	4.31	37.75
normalized	3.5	best-first	accuracy	0.3	SVM	0	3	81.7	95.69	83.22	4.31	37.75
normalized	3.5	best-first	accuracy	0	RF	3	3	83.61	97.83	83.5	2.17	44.5
normalized	3.5	best-first	accuracy	0.3	RF	3	3	82.59	99.15	82.5	0.85	21.5
normalized	3.5	best-first	accuracy	0	SVM	3	3	82.26	98.33	82.55	1.67	24.08
normalized	3.5	best-first	accuracy	0.3	SVM	3	3	82.1	98.86	82.35	1.14	15