Detecting User Engagement Using Mouse Tracking Data

Prifysgol Abertawe
Swansea University

David Saunders (910995)

Department of Computer Science
Swansea University

This dissertation is submitted for the degree of
Master of Science

September 2020

Declaration

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

David Saunders
September 2020

Statement 1

This dissertation is the result of my own independent work/investigation, except where otherwise stated. Other sources are acknowledged by giving explicit references. A bibliography is appended.

David Saunders
September 2020

Statement 2

I hereby give consent for my dissertation, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

David Saunders
September 2020

Abstract

This project explores the use of Hidden Markov Models to determine if users are paying attention during a crowdsourced study. The users are split into two distinct groups, online crowdsourced users, and lab study users. It is proposed that the lab study users were paying attention, and the crowdsourced users may or may not be paying attention. By using observed data recorded from user's mouse cursor it is possible to model both groups interaction with the system with separate Hidden Markov Models. Some user's interactions seem misplaced compared to their groups. These are reclassified with the aim of changing the groups of users from lab study or crowdsourced, to users paying attention and not paying attention. It was found that potentially only 9.8% of the crowdsourced users were paying attention, which highlights the known inaccuracies of crowdsourcing data.

Contents

List of Figures

List of Tables

1 Motivation

Crowdsourcing is a method of collecting responses from tasks that require human intelligence to complete. One such example of a crowdsourcing service is Amazon's Mechanical Turk [1]. A common use for this technology is to label data for use in training for machine learning algorithms [2]. They can also provide a cheap, scalable method for scientists to gather responses in research [3].

The use of gathering responses using Amazon's Mechanical Turk and other crowdsourcing alternatives are becoming prevalent across disciplines. It is commonly used in conducting clinical research [4] and it is estimated that almost half of all cognitive science research involves the use of crowdsourcing services to collect data samples [5]. The creation of an easy to implement method to measure user engagement would massively help researchers to increase reliability of their research.

The level of user engagement, attention, and low-quality responses can all be issues when gathering data from participants with distributed approaches [6]. The primary motivation of this project is to develop a system of potentially identifying if a crowdsourced user is paying attention during a task.

Research has been conducted on how the accuracy and attention of crowdsourced tasks can be increased. Methods such as offering financial incentives [7] and engaging a user's curiosity [8] have been found to motivate workers into performing better at crowdsourced tasks. Despite the research there is still debate as to which method is superior. If user engagement can be effectively identified, then the best method of ensuring user engagement could be found using this method.

Measuring user engagement is well studied in the field of web analytics [9]. However, the existing methods of reporting and analysing website data cannot be easily applied to crowdsourced tasks. Characteristics such as session duration and customer satisfaction are used as proxies for engagement. However, a longer session doesn't necessarily mean a more engaged user and customer satisfaction is not applicable for crowdsourced tasks. Therefore, existing solutions will not work and now methods must be evolved.

1.1 Background

The work in this project is built on data gathered from a previous study; *"A comparison of user interactions in lab and crowdsourced studies"* [10].

The study was centred in the field of data visualisation. The paper explored whether an alternative more detailed interface could be used help improve a user's effectiveness at completing a task. The task simulated an investment scenario, the aim was to maximise the return of a collection of stocks over several iterations.

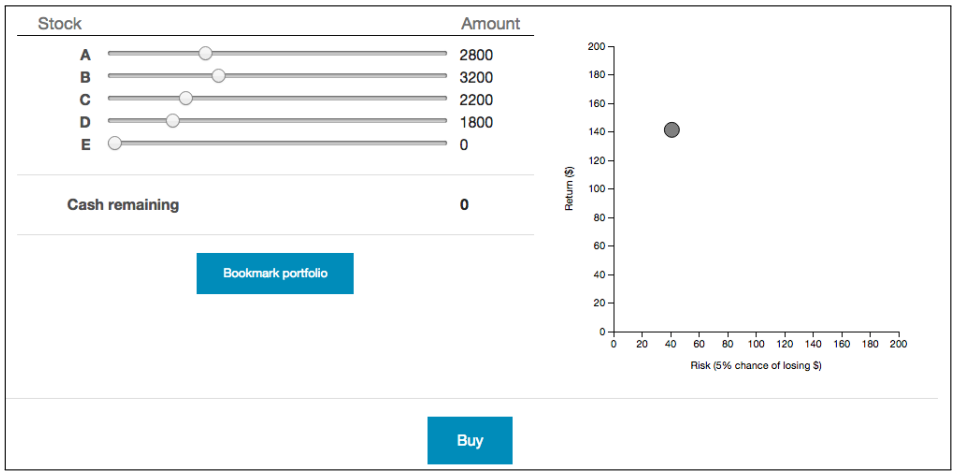


Figure 1: The main interface of the lab study, with the risk and return of stocks to the right and allocation sliders to the left.

Figure 1 shows the main interface of the lab study, with the risk and return of stocks to the right and allocation sliders to the left. Users were asked to invest their money by putting it into varying amounts of stocks. The 5 stocks were labelled from A to E. Depending on the makeup of the user's stock portfolio, the plot would update showing the return and risk of their portfolio. They were then asked to confirm their choice by buying these stocks. This process was repeated for 5 steps, where after each step the user's portfolio value with increase based on their previous investment decisions.

Figure 2 shows that the sliders representing each stock are split into two separate sliders, showing the return and the risk of the stock. Additionally, the sliders are no longer one dimensional, both sliders are two dimensional line charts. The assumption of the study was that these extra visualisations would help users to maximise the value of their portfolio.

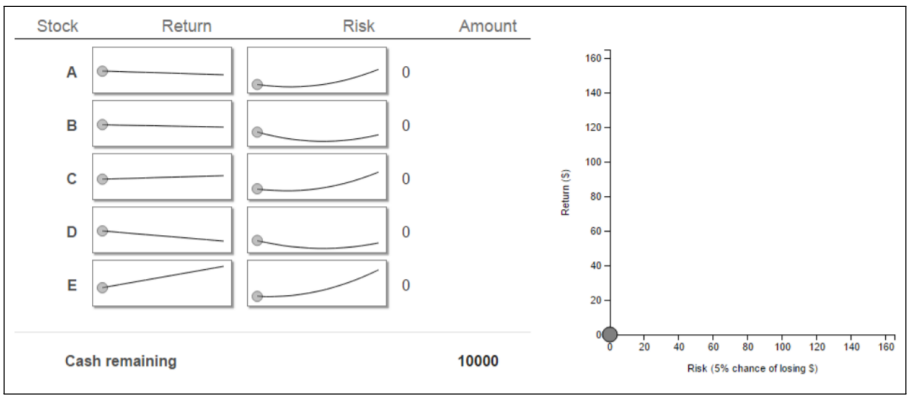


Figure 2: The alternative user interface of the lab study.

Participants from the lab study were heavily monitored to ensure they were engaged, focusing on the task. The crowdsourced participants were not monitored so it is unknown if they were engaged with the task, without further analysis.

Table 1: Data collection methods used in the study [10].

Data collection method	Number of participants	Were participants paying attention?
Lab study	18	Yes
Crowdsourced task	370	Unknown

Table 1 shows the two different ways in which data was collected. Only 4.6% of the data was gathered in person, with the most of responses being crowdsourced. The difference in number of participants shows one of the reasons crowdsourcing is popular. It is much easier to crowdsource responses than it is to organise an in-person lab study.

The study was inconclusive in proving that the alternative interface design had a meaningful effect on the success of a user's portfolio. Therefore, for the purpose of this project the distinction of which interface a participant was using is ignored. Additionally the rationale is that regardless of which variant of the interface was being used the participant would be paying attention. There is likely to have crowdsourced users who are engaged and some that are not engaged regardless of which interface is used. While there will be differences in users mouse data depending on the interface, I hypothesise that the difference will not be as large as the difference between users that are engaged or not engaged.

1.2 Initial attempt

One naive assumption might be that a users attention level might be inferred from the time taken to complete the tasks. Exploration of the data showed that the division of users may not be so straightforward, shown in Figure 3.

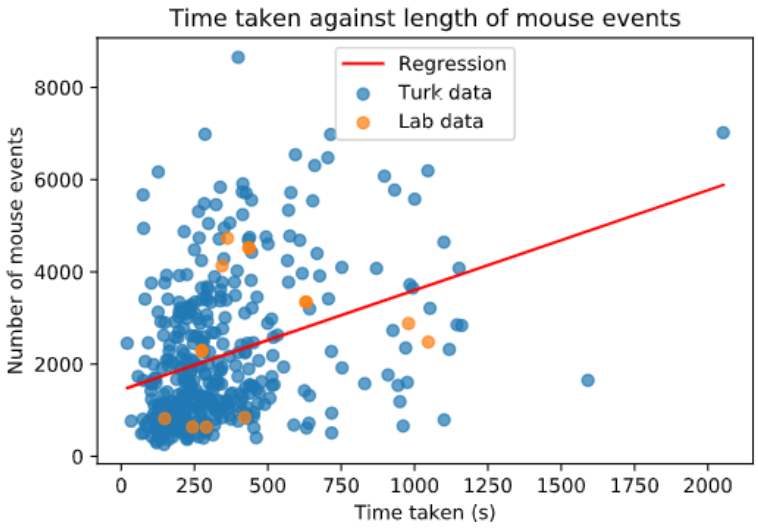


Figure 3: Scatterplot of users.

The lab and turk user classes cannot be separated by a simple metric of their data, such as length of events sequence and time to complete task. This eliminates the option of using distance based machine learning approaches such as K Nearest Neighbours or a Support Vector Machine as there is no relationship between such simple features. It should be noted that length of events sequence and time are correlated with a Pearson's correlation of 0.344.

1.3 Contributions

The contributions of this paper can be summarised as follows:

1. The development of a system to model a study participants mouse data, and to detect outliers from groups of users.
2. The creation of visualisations to explore participants mouse data.
3. The creation of methods of generating new data given small amounts of existing data.

2 Related Work

In this section I will discuss the related work to my project. This will include more general research in the fields of tracking user's mouse data and detecting a user's engagement. Works about crowdsourcing data are found which are key in understanding the background of this project. Machine learning techniques, and previous projects that have the possibility of being used in this context are also researched.

2.1 Eye tracking

Non-verbal information such as eye tracking may be used to detect user's level of engagement [11]. Vision is one of the most powerful human senses, so it has the potential to give a good measure of user engagement. The methodology of eye tracking is that we move our eyes to focus on particular areas that we want to see in more detail and divert our attention to that area [12]. Thus, tracking a user's gaze can provide insight into which part of a system they're engaged with, and how much so.

Figure 4: Eye tracking visualisation showing popular locations of users eyes on a webpage. Red areas are more frequently looked at than blue areas [13].

Eye tracking data can be used to show interface elements that users focus

their attention on, shown in Figure 4. From this data researchers were able to predict the amount of attention elements of the page would receive. By observing what parts of an interface users are interacting with we can determine what a user is engaging with [13].

Eye-tracking has been used, and found success novel applications such as recording the engagement of users when playing a game. Tracking users eye movements helped game designers understand how users can recognise interactable game objects and could be used to investigate problematic game design issues [14].

Eivazi and Bednarik extracted features from a users eye tracking data to determine their cognitive state in a problem solving exercise [15]. Features such as "mean fixation duration" and "total path distances" were engineered, and users were split into classes based on their performance. Given a user feature set and their performance class it was able to classify their cognitive state during the task with a 87.5% accuracy with a support vector machine.

Szafir and Mutlu identified a plethora of verbal and non-verbal behavioural cues used by teachers in an educational setting, with gaze being identified as one of them [16]. The behavioural cues could not be recorded directly by a computer, instead EEG signals measured from a headset were used to measure engagement.

Eye tracking is not however a perfect solution and its limitations have been well documented. Tracking subjects eyes with a good degree of accuracy requires the use of expensive, intrusive equipment that frequently needed recalibrating [17].

2.2 Mouse cursor tracking

Research has also found that there is a correlation between a user's gaze and their cursor position. The position can be considered a "poor man's eye tracker" as it has been found that eye gaze match mouse position 69% of the time [18]. Mouse movement data can be collected without the drawbacks of eye tracking and with more automatic methods, meaning more data can be collected, and on a larger scale [19]. Therefore, it can be said that mouse data can be used as a good alternative to eye tracking data.

By using mouse data, it is possible to unobtrusively record a user's normal use of a web browser without disturbing their experience [20]. It has also been found that users tend to follow the text they are reading with the mouse cursor

[21]. It can be determined what paragraph of a page was being read with an accuracy of 79% by using mouse cursor data [22].

Other methodologies explored ways of classifying user engagement from eye and cursor data, however it is also possible to predict user's attention and user frustration in complex webpages [23]. Not all studies agree that mouse cursor is always a good approximation for eye data. Hauger et al found that distinct cursor behaviour exists depending on the task, and that the relationship between eye gaze and mouse position is more nuanced than measuring only mouse data [24].

2.3 Crowdsourcing Cheating

The inaccuracies of responses when crowdsourcing results is a well-established issue. Users are paid per job complete, giving them incentives to complete tasks as fast as possible to maximise their income. Workers speed up task by submitting incorrect results, or by failing to perform the task correctly [25]. These workers can be labelled as cheaters, as their contributions are unusable.

The crowdsourcing platform Crowdflower created an integrated method of detecting cheating to address this issue. This method was used effectively when using human workers to help label datasets [26]. Tasks for which the answer is known are periodicity asked to the user, and the accuracy is measured. Once a user has completed a number of tasks over a threshold value then their accuracy on the known tasks is assessed. Of this accuracy is deemed to be unacceptable then the results from that user are ignored, and they are prohibited from contributing any more. This system has the drawback of taking more time and being more expensive than with no cheat detection method. Out of the 15 unique users that completed some of the tasks 9 of them where labelled as cheaters, presumably using scripts to randomly select answers. This allowed them to create a much larger volume of responses than any legitimate users, with 91% of the results coming from cheaters.

Crowdsourced tasks can be classified as Closed Class tasks and Open Class tasks, with each having unique cheating approaches [27]. Closed class questions are more common, requiring workers to choose an answer from a predefined list. This includes check boxes, buttons, multiple choice questions, and importantly to this project, sliders. Randomly picking answers is the most common cheating attempt, this can be detected by comparing a suspected users responses with a user known to not be cheating or comparing with all other workers answers.

Open class questions are much easier to cheat, these involve giving a user much more freedom to complete a task. Typically, workers may be given blank text fields to fill in, or blank canvas' to draw on. Standard cheating approaches involve leaving the fields empty, or repeatedly entering the same text. These approaches can be easily detected. More advanced attacks copy domain specific text copied from the internet, which is difficult to detect.

2.4 Spam detection

2.5 Semi Supervised Learning

This will be a key aspect of the project as we only have definitive labels for part of our data. This reflects the challenges of real-world data, where the quantity of unlabelled data is much greater than labelled data.

Semi-supervised learning is the study of combining both labelled and unlabelled data to improve machine learning techniques [28]. One of the most popular semi-supervised learning techniques are Semi-Supervised Support Vector Machines (S3VMs).

Reference [29] as it looks to be massively influential with 4,000 google scholar citations.

The data used in this dissertation is labelled. All data belongs to 2 classes, a crowdsourced turk user class, or a lab study user class. However, on the other hand the data is not labelled for the task I would like to explore. The goal of this project is to identity which users were paying attention. We have assumed that we can infer that lab study users will be paying attention, so we can say that those samples are labelled. However, the rest of the samples we have which are all of the online data are unlabelled. Therefore, this is a kind of semi supervised learning problem where we only have a small percentage of our samples labelled, and only confident labels for one class.

Semi supervised learning has already been applied successfully to the field of user engagement. During a 2017 study video data of secondary school children was recorded during a set of tasks, with the aim of classifying pupils between the states of 'Engaged' and 'Disengaged' [30]. Rather than annotate all data by hand which would be very time intensive, 1000 frames of video were selected at random and hand labelled. This led to a split of data labelled 563 engaged and 437 disengaged. AU facial recognition features were extracted from the pupils faces in the video frames, and important features decided with a Principal

Component Analysis method to keep features totalling 95% of the variance.

Their chosen method of semi supervised learning is a safe semi-supervised support vector machine. This variant of a semi supervised support vector machine finds a better separator when there is a small volume of labelled data, but a large volume of unlabelled data. The SSL model was trained with numbers of labelled data from 50 to 500, and an equal number of samples from both classes. The AUC metric ranged from 0.65 to 0.733 depending on the quantity of labelled data used, outperforming a traditional SVM.

Unfortunately, as shown in figure 3 the data here cannot be spatially separated and therefore any SVM would not be successful on the data. It is still useful however to see how semi-supervised learning has been applied to the field of detecting user engagement and how it has been successful.

2.6 Hidden Markov Models

A Hidden Markov Model will be the main statistical modelling technique used in this dissertation. In this section I will give an overview of what they are, and how they have been used in related projects. A Hidden Markov Model (HMM) is a statistical modelling technique based on the Markovian process. It assumes that the states of the system follow the rules for a Markov process, that is that every state depends only on the previous state of the system. It assumes that we cannot observe a system directly, but only see observations from it.

HMMs can be defined with the parameters (Q, A, O, B, π) [31]. Q is the set of hidden states of the system, the cardinality of which is normally selected empirically [32]. A is a transition probability matrix, a square matrix of shape $|Q| \times |Q|$. This matrix consists of the probabilities of transitioning from one state to another with each row and column representing a state in Q , and the main diagonal of the matrix representing a self-transition. O is the sequence of observations from the system. B is the emission probabilities of the model, these give the probabilities of each observation coming from each state. π gives the initial probability distributions of the system. It is a set with the same cardinality as Q, where each item in π relates to the probability of the model starting in a specific state.

HMMs have had success in modelling user behaviour when completing a simple task [32].

Paper Tom send me about HMM for text classification that I might be able to use [33].

Paper claims to use HMM for spam detection, I think they actually just use it to detect misspellings of words or something which is used by spam to hide from filters [34]. Probably could find a better spam detecting HMM, I just like how this almost does something different from the title, which my diss will end up doing.

This paper was recommended by someone on stack overflow as an old influential paper in the field with tens of thousands of references [35]. Called a tutorial on HMM and it's so old so original source. Would definitely be good to reference if I include any of the mathematics behind HMMs.

Someone's dissertation on the topic of generating synthetic data with HMMs. This can be a good way to create synthetic data [36].

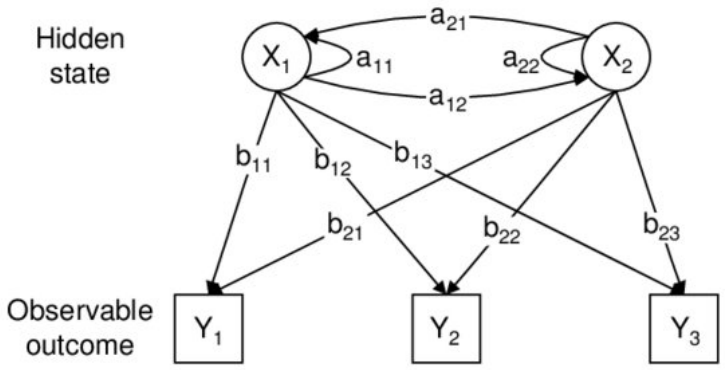


Figure 5: Diagram of an example Markov Chain model [37].

Figure 5 shows the states of a Hidden Markov Model of my system with states A-E representing sliders 1-5 and state F representing an html element. In the top right we can see a possible transition matrix of our system.

HMMs are typically used with sequential data such as time series data.

3 Methodology

The overall methodology of this dissertation is to conduct exploratory research.

The assumption of this project is that labs are paying attention and turks are not. Therefore any outliers from the turk data that appear to be more similar to the lab data will be examined. The outliers can then be classified as potentially paying more attention than his peers

This study (ref) says that only 10% of all people / turk users pay attention during a task. We will look at the 10% (30ish) of the turk data that looks like it is lab data and day that they were paying attention. This is just the assumption we have made for the project, unfortunately the dataset isn't extensive enough for us to fully test this hypothesis.

4 Data Pipeline

When planning and completing this project many decisions were made about the steps taken to convert the raw data to a finished project. This section may act as an overview of the project, detailing the different sections of work, what they may contain, and the order in which they will be completed.

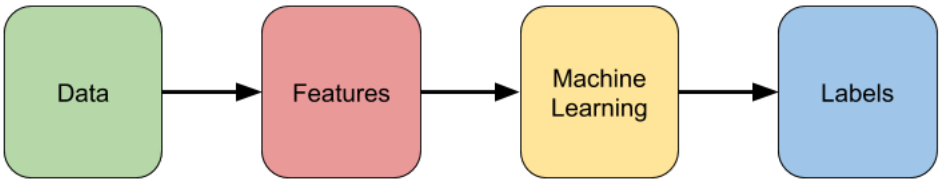


Figure 6: Diagram of the Data Pipeline of the project.

4.1 Data

The first component of the project has been completed, and data has been extracted form JSON format to a csv format.

Table 2: The first 5 records of results of the crowdsourced task.

event_type	target	time	x	y	step	turkId
mousedown	alloc-slider-1	0	477	405	1	A35YFAFWP33C70
mouseup	alloc-slider-1	0.111	478	405	1	A35YFAFWP33C70
click	alloc-slider-1	0.111	478	405	1	A35YFAFWP33C70
mousedown	alloc-slider-1	1.516	479	405	1	A35YFAFWP33C70
mousedirchange	alloc-slider-1	2.395	543	403	1	A35YFAFWP33C70
mousedirchange	alloc-slider-1	3.161	594	402	1	A35YFAFWP33C70
mouseup	alloc-slider-1	5.048	514	407	1	A35YFAFWP33C70
click	alloc-slider-1	5.048	514	407	1	A35YFAFWP33C70
mousedown	alloc-slider-2	5.461	494	441	1	A35YFAFWP33C70
mouseup	alloc-slider-2	5.513	494	441	1	A35YFAFWP33C70

Table 2 shows us the features of the data. The lab study data and the crowdsourced data have the same schema, except lab results have a different ID field.

The target field shows us which element in Fig 1 a participant is interacting with and event_type details the type of interaction. Time field shows the time taken in seconds since the first recorded mouse event. We can hypothesise that participants with a shorter time paid less attention than a participant who took much longer, thinking about their actions more. The x and y fields show the location of the mouse and step shows which stage of the task, from 1 to 5, a participant was in.

4.1.1 Data Manipulation

Here I will summarise what I have done to the data. As previously mentioned the data was gathered through a lab study and an online crowdsourced study. The purpose of the online study was to gather a larger amount of data that was possible to do so in person. Data was recorded in a JSON format with lots of irrelevant and duplicate data relating to the users background and not their mouse movements. JSON is a form of unstructured data, meaning it is not fully structures but has some organisation to it [38]. To use the data in this project it first had to be processed into structured data. This was challenging and time consuming as Python's JSON library was unable to directly convert the data. This was because the mouse events were stored as a nested JSON dictionary and there were additionally errors with the way the data was logged causing it to be invalid JSON. After these challenges were addressed the data was converted and saved to a Pandas DataFrame, and then to a csv with over 100,000 lines. Errors were found in some of the data and so the final number of usable data is less than the figures shown in 2. Example errors include missing mouse events, or time to complete task being recorded incorrectly as 1.4×10^{12} seconds or $31,688$ years, obviously incorrect.

4.2 Features

This part of the pipeline refers to what features I am going to extract from the data. Features of data can be defined as "a measurable piece of data that can be used for analysis" [39]. These will consist of both raw and created features, but what do I mean by this? Raw features will consist of the number of mouse

events recorded, while a created feature would have to be extracted from the data.

One such feature recorded in the data is mouse target. This refers to the HTML element that a users mouse was over at any given time. As shown in Figure 1 the most prominent part of the interface that users will interact with is the 5 stock allocation sliders. It is hypothesised that users who are paying attention may spend more time fine tuning their stock allocation and therefore would have more mouse events with the sliders as the target. An attempt to pick up on different parts of the interface was achieved including which element of the stocks plot a user was interacting with. However many html elements were ambiguously named and so only the sliders are named from 1 to 5.

The html target will be used as a feature in this project. Additionally a users mouse location over time may provide useful insight into how they completed the task therefore it will also be used. This feature was even more challenging to engineer as the data needs to be altered to be useful time series of data. Ideally the data should be given as a multivariate time series which is a sequence of numerical vectors [40]. The data should be in the form

$$\langle (t_0, (x_0, y_0)),\ (t_1, (x_1, y_1)) \dots (t_n, (x_n, y_n)) \rangle$$

where t is the time in seconds, given from the first timestamp t_0 in whole seconds to $t = n$. The tuple (x, y) refers to the horizontal and vertical location of the users mouse cursor.

4.3 Machine Learning

Once we have insightful features from the data, we can consider what machine learning algorithm would be most appropriate to use on the data. This will obviously be highly dependent on what form the final features are in. For example if the features are numerical values such as time taken to complete task and number of mouse events then an algorithm such as a Support Vector Machine could be a good choice. If the data is in the form of sequential data such as a sequence of all mouse events then alternative techniques a Hidden Markov Model or neural network would've appropriate choices [40]. Variations of the traditional neural network such as Recurrent Neural Networks (RNNs) or Long Short Term Memory (LSTM) Networks are known to perform better with sequential data [41]. If the features output was an image such as a trace

of mouse position over time, then a Convolutional Neural Network could be a suitable choice as they're designed for image data [42]. An issue with any neural network approach would be their tendency to overfit the data. That is when an overly complicated model tries to model relationships that are a result of noise in messy data. There is only a limited amount of training data for this project, and so a high risk of this happening. Despite techniques such as dropout [43] existing to address this issue I do not think any form of neural network would be appropriate for this task.

It is likely that text classification algorithms will be used when comparing the html targets of mouse events. Comparing counts of n-grams can be done with algorithms such as probabilistic Naive Bayes classifiers. Other suitable classification algorithms include proximity-based k-nearest neighbours and ensemble decision tree learners [44].

4.4 Labels

Lastly an important section of the pipeline, as it reflects the final outputs of the system. Labels will refer to which users are classified as paying attention and which users are not. A key aspect of this project will be semi-supervised learning. That is we have some data points we can confirm were paying attention, and others where they're level of attention was questionable. Once we have an algorithm that can classify some users as paying attention or not we can rerun the algorithm with these preliminary outputs as new training data. If this is done recursively then we can end up with a system that can split all data points into the 2 classes, perhaps with a degree of confidence given as a percentage.

5 Final Implementation

As defined previously a HMM consists of the 5 parameters (Q, A, O, B, π) [31]. In the case of this project, the hidden states Q would relate to some aspects of a user's thought processes as that is the behaviour that is being modelled. The observations O relate to the features as defined in the features section. Features used are the users mouse location over time and sequence of interface elements. The parameters (A, B, π) are all learned during training, so do not need to be specified [45].

Figure 7 shows the design of the main model used in this dissertation. It was decided to use only the series of mouse events and ignored the timestamps of

the event. This is because there was no previous correlation discovered between a user's total time taken or the number of mouse events. Therefore, there is no reason to assume that using a time series would be any more accurate than a non-time series of data. The sequence of mouse events should still hold the detail that time data would.

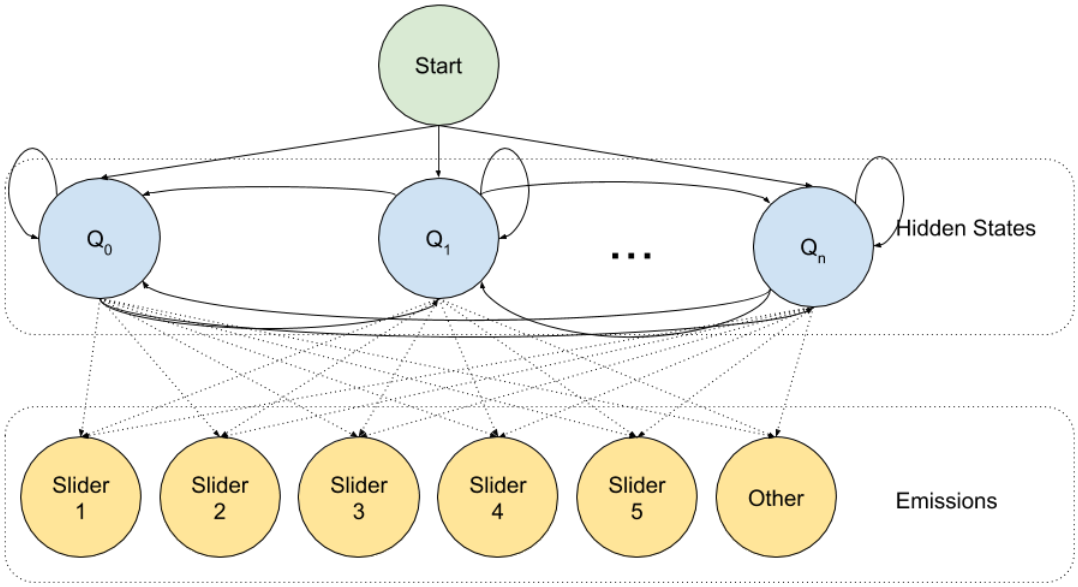


Figure 7: Diagram showing the states of the designed Hidden Markov Model. The emissions O are shown to be the mouse targets. The number of hidden states Q are not specified as that was decided in Figure 8. (A, B, π) are also not shown as they are calculated and not specified by me.

With Bag-Of-Words models the order of items is ignored and only takes into consideration the frequency of each item. This is a different approach, here the order is the most important thing.

To implement a Hidden Markov Model the python library HMMLearn was used [46]. The MultinomialHMM model was used as the emissions are discrete, representing the different interface targets. The sequence data used to train the models was the order of mouse targets for each worker. The temporal aspect of the data was ignored, but earlier research showed that number of mouse events and time were heavily correlated. Therefore using just the order of mouse events should capture the temporal aspect of the data.

When implementing a Hidden Markov model on the actual data there are lots of parameters that must be considered. Given a sequence of input data the HMM model can calculate the parameters of the model such as the transition matrix, emission probabilities, and initial probability distributions by using the Viterbi algorithm [46]. The most prominent parameter that must be given is the number of hidden states to use.

The paper [47] addresses this issue on similar data. The authors use a HMM to model animal movement behaviour, where they hope the hidden states would roughly relate to the behaviour of the animal. For example in this context a 2 state HMM may relate to "foraging/resting" and "travelling". Humans can be more complex than animals but we can think of potential states of being "inquisitive" and "bored". They trained models with 2, 3, 4, and 5 hidden states and compared the criterion's of AIC, BIC, ICL. Similarity to evaluate the Lab and Turk models, several models were trained with number of hidden states ranging from 1 to 11. Generally, an increase in states should increase the effectiveness, however it will also take longer to train. All models used for this comparison was trained to 50 iterations for a fair comparison.

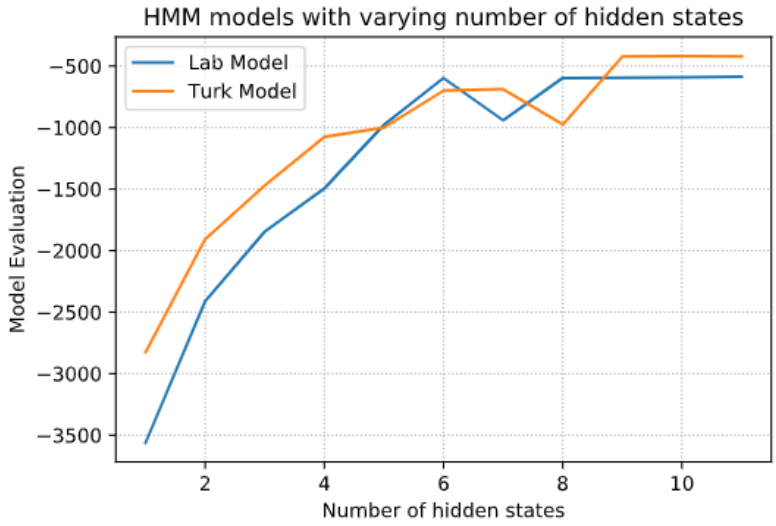


Figure 8: Line plot of evaluation of different HMM architectures.

Figure 8 shows a different criterion of the total log likelihood of the training data of the model. This value is normalised by dividing by the number of data samples, to give a per-sample total likelihood. The evaluation of each HMM

model does fluctuate based on the random state used, which is why there are drops in the evaluation criterion even when more hidden states are used. We can see that the criterion for both models appeared to stop improving after 9 states. Therefore a 10 state model was used to ensure that the models could be as accurate as possible while retaining reasonable complexity.

6 Repeated Experiments

This section will give an overview of the work I have completed during this project. Each sub section offers a unique approach to tackling the problem. The different methods were tried sequentially, and the pipeline had to be amended for each one as the input requirements changed.

The approach of this project was to try many different methods of looking through, classifying, and predicting attentiveness of this data. These experiments enabled me to get a better understanding of the data, and to explore which methods would be most appropriate. While these results did not end up directly contributing to the main goal of this paper, they were important steppingstones of the project.

Before any users were reclassified based on their engagement an alternative aim was established. This was to create a method of identifying whether a user was from the lab study or if they were a crowdsourced turk user based on some input data. While this is not the ambition of this dissertation, it would reveal interesting information about the two user groups. If it was trivial to tell these groups apart then it would suggest that there may have been an error in how the data was recorded, or some other unintentional feature a classifier may be identifying. If any machine learning method was unable to classify a given user then it may suggest that there is little to no difference between the users data. This could additionally indicate that there is no link between a users mouse data and their engagement, which would be a huge and surprising discovery.

6.1 SVM separation method

This section provided the initial exploration into the problem. These experiments attempted to separate the data samples for crowdsourced turker users, and lab participants by plotting samples in space. The aim is that this might reveal information about the features of different users. If there was a clear distinction between the classes then we could potentially reclassify some points

based on their euclidean distance to one another.

A Support Vector Machine (SVM) is a common supervised machine learning technique [48]. Here we look at if a simple hyperplane could separate the data when looking at the simple features of the number of mouse events and the total time taken to do the task.

It was found that a linear, or non-linear classification with a kernel trick was unable to separate the data. However as shown in Figure 3, the data does not appear to be linearly separable. Regardless this method was still attempted.

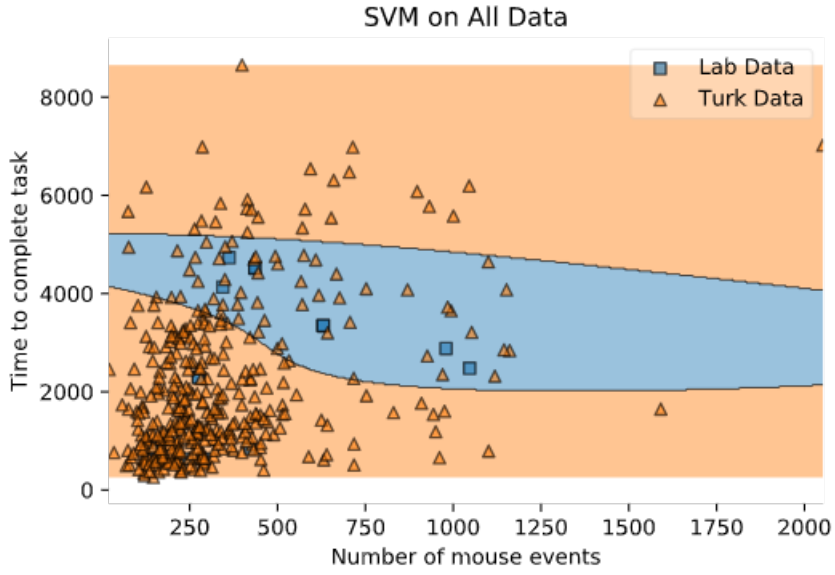


Figure 9: Plot of the SVM, showing the decision region boundary.

Figure 9 shows the unsuccessful attempts of this method and shows the sheer imbalance of the dataset. Only 53% of the lab data was within the relating decision boundary, whereas 86% of the turk data was in its respective region. This highlights how this method will be ineffective on the dataset. While the SVM has created this central band of lab data, this region doesn't accurately model all the data.

Additionally, this method highlights a reoccurring problem of this project, the unbalanced data classes. This can make the accuracy of the model high, even if it misclassified most data points.

6.2 Text Classification

This section of experiments is inspired by text classification algorithms. Text mining is becoming increasingly popular, due to the large increases in available text data from the web and social media. Such data is typically unstructured and contains very large amounts of information. Such data is also sparse and high dimensional [49]. The high dimensionality of text data stems from the large lexicon of words a document may contain, just the English language has over 170,000 words [50]. Text data is usually sparse because a given document is unlikely to contain anywhere near that many unique words.

The users mouse data used in this project can be represented as an abnormal word document. We can consider a mouse target as a "word", and a "text document" as a single users sequence of mouse targets. This data will have a very limited lexicon of only 6 "words", the sliders from 1 to 5 and other. Therefore, this data will have much lower dimensionality than typical text data. Additionally, every user has interacted with every target, therefore the data is not sparse but dense.

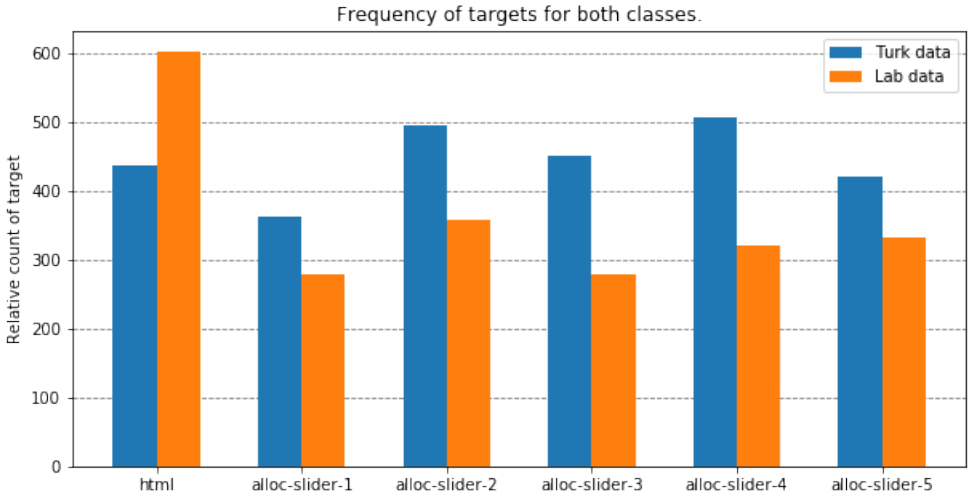


Figure 10: Histogram of targets.

Figure 10 shows the relative count of each of the targets for the user groups. The relative count is calculated by dividing the count of the targets by the number of users, this is so the numbers are comparable otherwise the crowdsourced data would be of a much larger quantity. We can see that lab study users seem

to perform more mouse events over the other html elements and not over one of the sliders. For lab users the most common sliders are 2, 5, then 4. For turk users they are 4, 2, then 3. The causes of these differences are unknown. Importantly they show that there are differences in the data from the classes, something that a classifier should be able to exploit.

One popular text classification method is with a Bag-Of-Words model, called such because the grammar and order of the words is ignored and only the counts of each word is used as a feature [51]. Grammar is meaningless in the context of mouse data, however the sequence of the words may hold crucial information regarding what class a user is in. Disregarding the sequence may be a big limitation of this method as we can imagine a user switching between sliders frequently may be very engaged in the task. Therefore disregarding the order of targets could be detrimental to the model. The Bag-Of-Words representation of the data can be used as input to a Naïve Bayes classifier [51]. A Naïve Bayes classifier predicts a class based on the probability of seeing the counts of words in a document. It can potentially be used here to classify users as lab study or crowdsourced based on their counts of targets. For example it may be found that crowdsourced turk users may have a higher lower counts for the later targets, potentially showing how they got tired of the task and stopped putting effort into it.

$$\begin{pmatrix} 322 & 430 & 464 & 308 & 230 & 1317 \\ 332 & 120 & 112 & 123 & 59 & 164 \\ 41 & 367 & 173 & 565 & 278 & 449 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

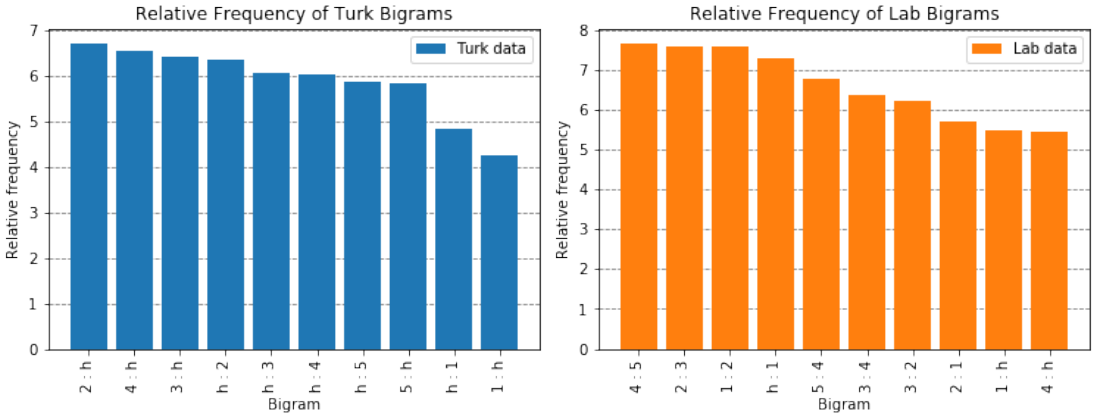
This matrix shows the first 3 rows of the bag of words representations of each users data. The matrix has 6 columns representing each of the mouse targets or "words".

This attempt of classification was a failure. The accuracy was 29.6% and F1 score of 22.8%.

6.2.1 N-Grams

The previous Bag-Of-Words method would be considered an unigram model of natural language processing. An n-grams is a series of n words, with unigrams being one word, and bigrams being two word pairs [52].

Different n-grams can be combined together to better understand the complexities of a text document. A mixture of unigrams, bigrams, and trigrams are used extract different levels of text complexity and perform well with document classification [44]. The same technique can be applied to this mouse dataset. Bigrams will be of interest as they show the transitions from one target to another, thus adding some sequential data to be used by the model. However higher n n-grams could reveal information about a users approach to the tasks. A user who has many n-grams of different sliders would mean they often changed between sliders, potentially looking for the optimum slider values.



(a) Turk Bigrams

(b) Lab Bigrams

Figure 11: Plots showing the relative frequency of bigrams for both groups.

Figure 11 shows the bigram differences of the crowdsourced turk data and the lab study data. By far the most frequent form of bigram was the same target followed by the same target. This indicates that if a user is using a slider their next target will most likely be the same slider, which intuitively makes sense. To address this issue bigrams leading to themselves were excluded from the visualisation. There are 36 unique bigrams, however only the top 10 most frequent bigrams are shown. We can see that the turk data is more likely to have a html element somewhere in the bigram, whereas lab users are more likely to have just have sliders. This is surprising as in Figure 10 it was shown that a html target was more common in the lab study data. The plots show how bigrams can be used to separate the data, it is hoped that this may help classification especially when combined with trigrams.

With a combination of unigrams, bigrams, and trigrams the matrix now

grown massively to have a total number of 222 columns.

However due to the problems there is still an accuracy of 94.9% and an F1 score of 48.6.

6.3 Imbalanced classes

Hidden Markov Models are generative models. They model the most likely output sequence [32].

As detailed previously HMMs can be used for classification as shown above, but they are also known as generative models [36]. This means they can be used to create new data. The above methods of Bag-Of-Words and N-grams failed due to the imbalances of classes.

There are many methods to address the issue of class imbalances. Class imbalances involve a minority class (in this case lab study data) and a majority class (in this case crowdsourced turk data). Two of the most popular solutions are to over-sample the minority class or to under-sample the majority class [53]. In there basic forms under-sampling effectively ignores data from the majority class, and over-sampling effectively duplicates existing data.

It was decided that under-sampling would not be appropriate as there is only a small amount of data, therefore as much data must be kept as possible. Over-sampling was a better solution however, to balance the classes the number of samples must still be increased by around 20 times as much. Trained intelligent undersampling methods can perform better but are more difficult to implement [53]. Therefore it was decided to use the existing Hidden Markov Model to generate new data. A trained HMMs can be used to generate new data points, to reduce the imbalances, and make the previous classification methods more effective.

A HMM can be used generate a probable sequence of n observed states, however to generate new samples we must decide what length n must be. Given n a walkthrough of the model is performed, the transitions between states then occur based on the probabilities within the transitional matrix. The idea of this section is to generate new lab study data so that there is the same quantity of lab study data and crowdsourced turk data. This generated data can be used to train the models used previously with the hope that this new data will address the issue of unbalanced data. The turk data and the generated data will be split into training and testing data as standard. However, the original lab data will not be used to train the model, this will be withheld to see if the models

trained from the generated data will be able classify the actual lab data.

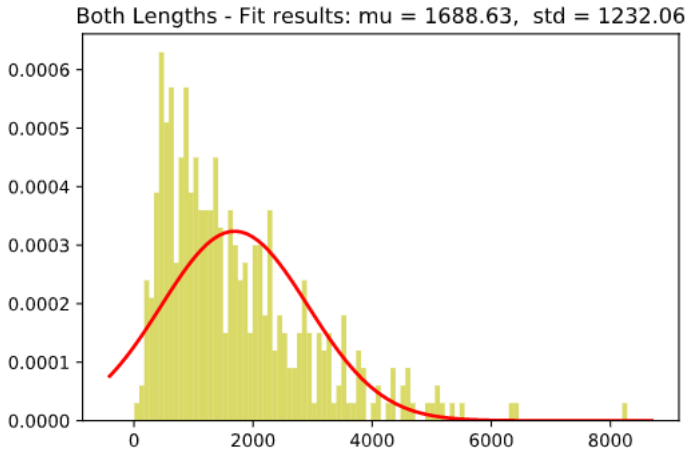


Figure 12: Histogram of length of mouse events sequences for both groups.

Figure 12 shows the frequency of different mouse events lengths. In red we can see a positively skewed normal distribution which matches the distribution fairly well. When determining n for the new sequence data, we will sample from a gaussian distribution with a mean of 1688 and a standard deviation of 1232. This should generate a realistic distribution of lengths, and when combined with the trained HMM model, a realistic sequence of data.

From this distribution 378 values were sampled. Each of the 378 values were then passed into the generative lab study HMM. This led to the creation of an equal number of crowdsourced data and synthetic lab data generated from the model.

6.3.1 Revisiting methods

With the generated lab study data we can now revisited some of the methods that failed due to imbalanced classes. The SVM classifier was unable to be revisited as the features used before were the time taken to complete the task and length of mouse events sequence. The generated data has a length, however there is no corresponding time feature. Despite the data not being suitable for a similar SVM the data was perfect for the previous Bag-Of-Words methods.

Using the Naive Bayes classifier with a Bag-Of-Words methods, there was improvement. The model was trained and tested with 5-fold cross validation which led to an accuracy and f1 score of 60% and 58% respectively. When the

bigram model was considered the accuracy increased by 1% to 61% and the F1 score remained the same. Additionally using the original lab study data as testing led to the same results from both models with the same confusion matrix.

$$\begin{pmatrix} 0 & 0 \\ 2 & 12 \end{pmatrix}$$

The confusion matrix from the models shows that of the 14 original lab study participants, only 2 would be misclassified as a crowdsourced participant. This shows that there is enough difference in the data for a classification algorithm to distinguish the points.

7 Results

Hidden Markov Models were trained on both lab study users mouse data and online crowdsourced mechanical turk mouse data. Then each users mouse data was evaluated by each HMM, and the log Likelihood of that sequence belonging to that model was recorded. This will tell use the likelihood that that sequence of mouse events belongs to either of the classes. If there are any points that appear to be outliers, then we can say that they were doing something differently to the majority of their peers. Outliers with a higher likelihood of belonging to the HMM trained on data from the other group then we can say that that outlier sample appears to belong to the wrong group. Or at least that outlier is doing something different to their peers, which may be a higher level of attention given to the task. These points were labelled Reclassified Lab data and Reclassified Turk Data.

7.1 Graphs of results

One way of attempting to classify users is to identify any users with a higher lab likelihood as being more similar to a lab user than a turk user. Figure 13 shows this result graphically. Any data point above the line $y = x$ has a higher likelihood of belonging to the lab model than the turk model. Therefore any turk data that is above this line can be reclassified as seeming to belong more with the lab data than the fellow turk datapoints. An interesting and surprising result is shown from the lab data. Not all of the lab data is shown to be belonging more strongly to the lab model than the turk model. 6 out of 14 (43%) of the lab data samples has been reclassified by the algorithm as being

more similar to the turk data.

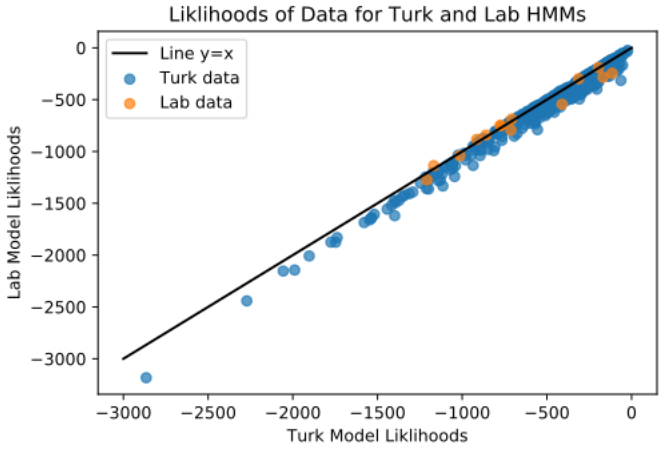


Figure 13: Scatterplot of users likelihood of belonging to either group.

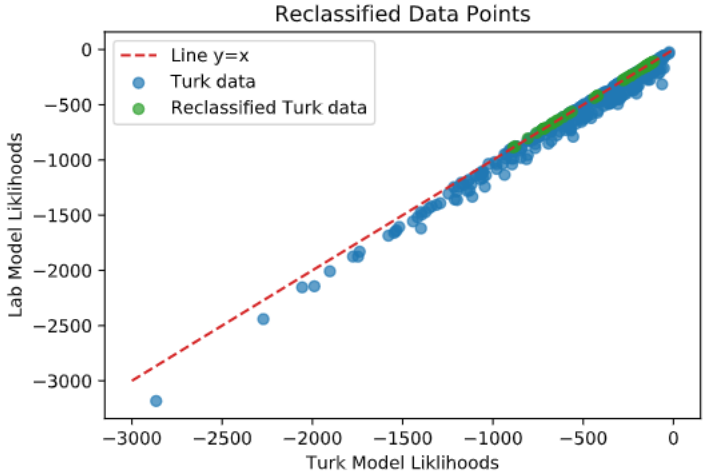


Figure 14: Scatterplot showing the turk users that have been reclassified as behaving more like lab users.

Figure 14 shows which of the turk users are more similar to the lab users, than other users from the same group. The plot shows the same datapoints and axis as in Figure 13, but the points above the line are recoloured green to show their difference from the actual lab data and normal turk data. 45 of the 361 (12.5%) turk users are above this threshold. Which seems like a reasonable

percentage of the population to be paying attention.

The exact distribution of these points change based on the initial random state when training the model.

Figure 3 showed a naive simple attempt to separate the classes of turk users and lab users. It was concluded that a spacial based technique such as a SVM would be unsuitable as the data didn't seem to form any patterns or clusters. Figure 15 shows how the reclassified datapoints are not linearly separable. This proves that my initial hypothesis was correct, and that data could not be reclassified purely by looking at simple feature of time taken and just number of mouse events.

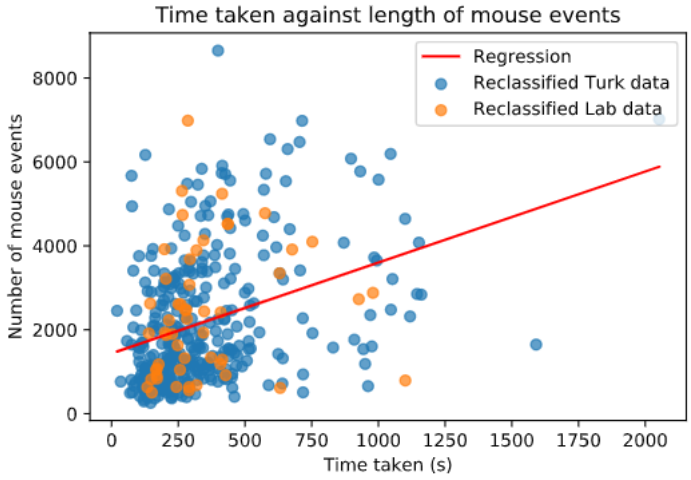
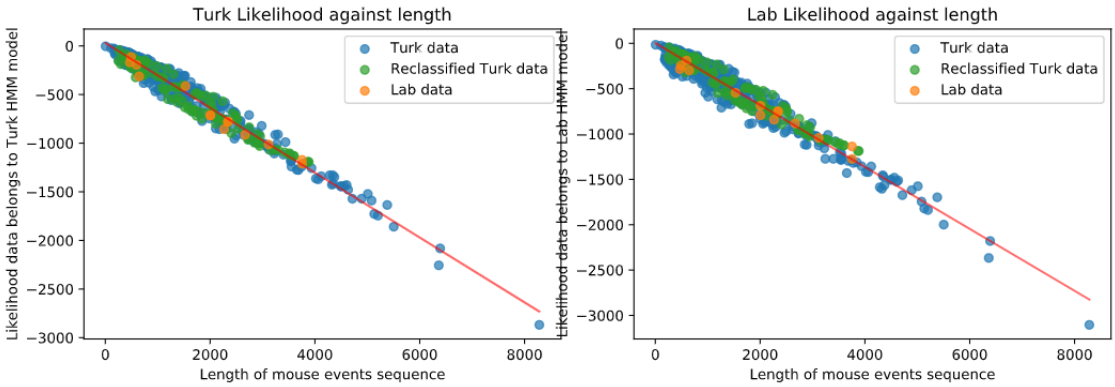


Figure 15: Plot showing features of the reclassified points.

7.2 Likelihood length correlation

Examination of the extreme points with the highest and lowest likelihoods revealed a potential correlation likelihoods and length of mouse events sequences. I decided to plot this data to understand if there was a link between these features.

Figure 16 show that the log likelihood from both models and length of a users mouse sequence are highly inversely correlated. The Lab model likelihood and length have a correlation of -0.982, and similarly the Turk model has a correlation of -0.976. A value less than -0.7 would indicate a strong negative correlation, therefore Figure 16 shows an extremely strong negative correlation



(a) Turk HMM Model

(b) Lab HMM Model

Figure 16: Plots showing correlation between likelihood of different models and the length of mouse events.

[54]. Such a strong correlation could indicate the models are doing nothing, other than looking at the length of a sequence. However looking at the reclassified Turk data we can see that there doesn't appear to be any correlation between length, and whether the data has been reclassified. Therefore such a strong correlation between the length and likelihood is of no concern.

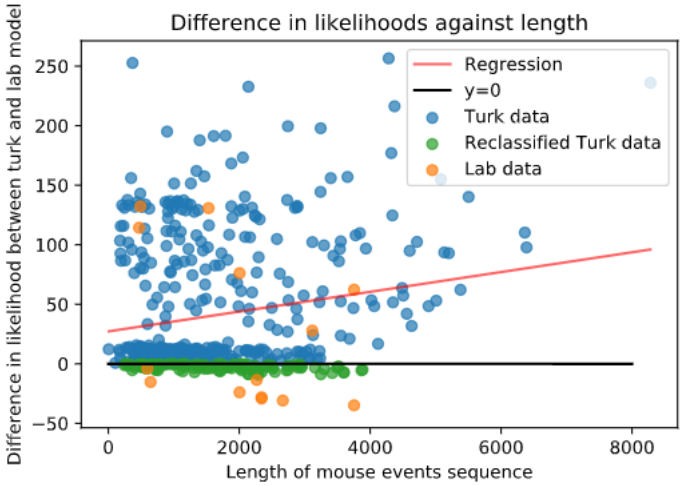


Figure 17: Plot showing difference in likelihood between the models against length of mouse events sequence.

Figure 17 removes the concern that length of mouse events sequence is the

only feature being used in the models. The difference in likelihoods and length of mouse events sequence has a correlation of 0.176, meaning the relationship between the features is non-existent.

7.3 Secondary Model

In order to confirm any findings, it was decided to attempt another experiment using similarly designed HMMs. The key difference is that these models will be trained with different data than the previous models. If models are created with different data, but they identify the same users as potentially belonging to the wrong group then it would be a confirmation that those users are indeed outliers. If the models predict a completely different group of users then it would indicate there is no consistency within the data and that the models may not be reflecting the actual processes.

The initial ideas was that confirmation models would be trained on multivariate time series as described in Subsection4.2 Features. However using a time series of data caused too much variance in the results, meaning that it could not be modelled by a Hidden Markov Model without access to more data. Therefore the sequence of mouse locations was used instead which can still provide interesting results as shown with the other model. Because of the different data this model will be a Hidden Markov Model with gaussian emissions as the multivariate inputs are continuous.

The reason that this data is only a secondary aspect of the dissertation is due to concerns with the quality of data. Particularly that the mouse location data does not seem to be calibrated correctly. Looking at the mouse cursor position for each position it would be expected that many of the cursor locations each user would be the same. For example, you would expect to see 5 distinct clusters of intensive mouse activity relating to the 5 allocation sliders. To end the task workers must press a "Buy" button to purchase their selected stocks. Therefore, the last recorded mouse event of all users should have the same position, that of the button. Unfortunately, this was not the case.

Figure 18 shows the issues with the mouse location data. Rather than the mouse events for the sliders having roughly the same there appears to be multiple groups with varying coordinates. The majority but not all of mouse down events happen within 2 groups with the same y coordinates. Additionally there is a massive range in mouse data. Most data falls below an X value of 800, and most are beneath a Y value of 1000 however there are still outliers to this. Figure 18

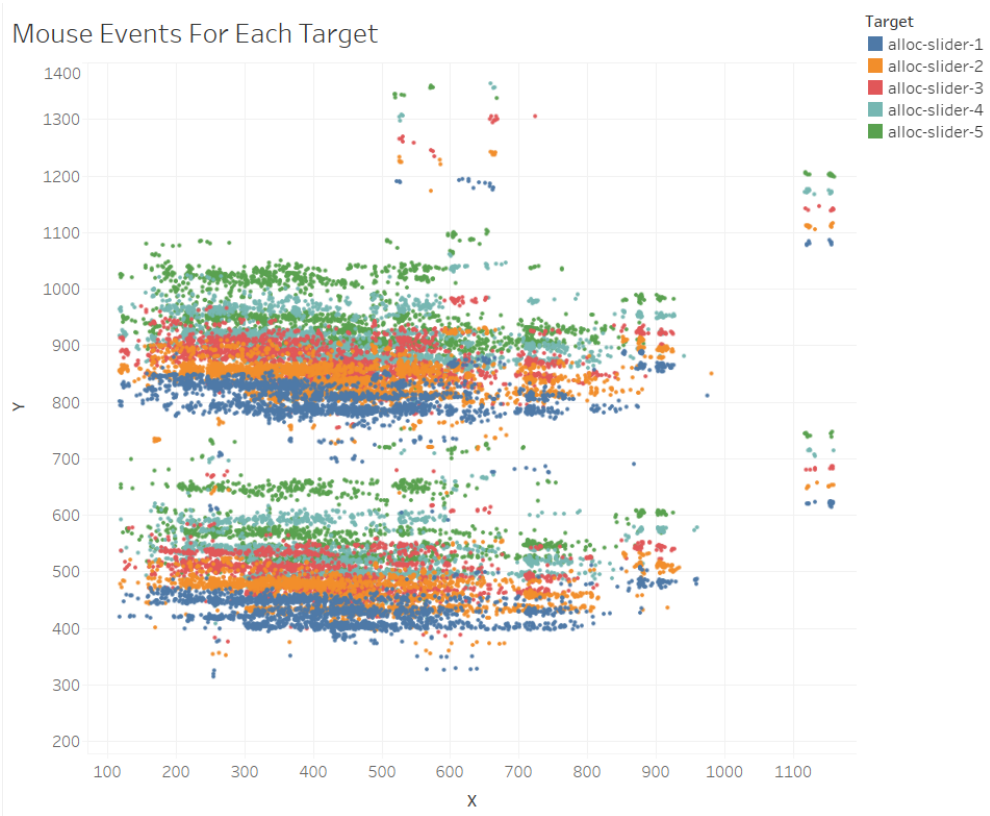


Figure 18: Scatterplot of mouse down event coordinates of all users mouse path data.

shows only the mouse down events on the slider targets, when considering the other mouse events and other targets the data was even more messy. The cause of this is unknown but is expected to be a result of the data recording process.

This messiness of this mouse data was not addressed. This is the reason that this method will only be used as a supporting secondary model. It is impossible to draw concrete conclusions from such messy data. This data may cause some of the users data to be unlikely to belong to any model as they're such a large outlier.

7.3.1 Results

Figure 19 shows how the new HMMs reclassified 80 of the crowdsourced turk users as belonging more to the lab study. It is worth noting that the likelihoods

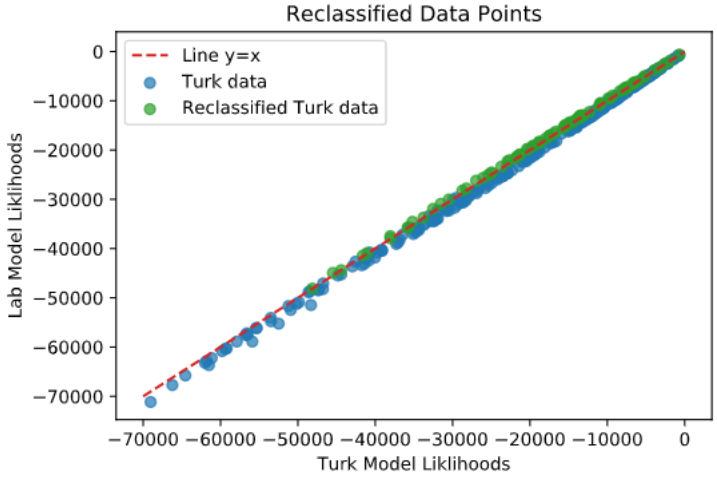


Figure 19: Graph of reclassified datapoints from spatial data HMMs.

of each user belonging to the model is much lower when the spatial data is used, compared to when the mouse target data was used. This is because there is much more variance in the continuous spatial data rather than the categorical target data. The original previous HMM classified 45 of the crowdsourced users. Taking an intersection of the sets of users will show us how consistent the results are. If there is a large intersection of users then the results are consistent with before, however if there is little to no intersection then it would indicate that there is no consistency between models.

$PreviousReclassifiedUsers \cap NewReclassifiedUsers$

There is an intersection of 12 users between the two sets of users. This indicates either that there is some consistency in the results. 27% of the originally reclassified users were identified by the secondary model. This is promising as it shows that a number of the reclassified users are shared between both methods. The inconsistency in the mouse location data identified could be causing the differences in reclassified users between the methods. Therefore I would predict that the results from the spatial models to be less accurate and representative of the real world processes that the models trained with mouse target data.

If we take the union of both sets of reclassified users we get a set of 200 reclassified users. This allows us to have a few different sets of users that we may reclassify as belonging to the wrong group, with different confidence levels. I am most confident in saying that the 12 users reclassified by both versions of

the models seem more similar to the lab study users than the other crowdsourced users. I am next most confident in the 120 users reclassified by the HMM trained with mouse target data. This is because data used in those models was more properly cleaned and had less variation in data. The users I am the least most confident in reclassifying is the union of users from both sets of models. This consists of 200 crowdsourced users which is 43% of the total population.

Table 3: The different sets of crowdsourced user paying attention with confidence.

Number of Users	Percent of Population	Identified By Models	Confidence Level
12	2.6%	Both Models	High
120	26.0%	Target Data model only	Medium
200	43.4%	Sequence Data model only	Low

Alternatively inverting this gives us users which we can assume are not engaged in the tasks, to different degrees. I am most confident in saying that 261 of the crowdsourced users not reclassified form the union of the models belong with their class and should not be reclassified. I have next most confidence in the 341 users that were identified with the mouse target data models. Lastly we have the 449 users not identified by the intersection of the models. I am least confident in saying that these users were not engaged in the task.

8 Future work

All the problems addressed in the conclusion have potential to be addressed in future work. As stated the main flaw of this project is that the label is not labelled for the task I am trying to perform. Therefore to fully evaluate any of these results would require further research with a properly labelled dataset.

It was hypothesised that a users mouse data would not be significantly different depending on which version of the study interface they were using. This hypothesis was based on the findings of the previous study, where the different interfaces had little to no impact on a users success at the investment scenario [10]. Perhaps an investigation into users mouse data might show more of a difference between the behaviours of users using both interfaces.

I would have liked to perform an additional lab study session where we could have generated more data for that class of participant. This would have helped alleviate the issue that the imbalance of classes caused. Unfortunately, this was

impossible due to time restraints and the interference of the global COVID-19 pandemic.

Another area of future work would be to see if any of the methods developed here could be applied to other similar datasets. A kaggle crowdflower dataset seems like an idea candidate [55]. While it is still not labelled with attention and non-attention, it does contain mouse data of crowdsourced users. Additionally the dataset contains the users results from 3 different tasks, a users success at a task may prove to be a reasonable proxy for engagement.

9 Conclusion

This dissertation aimed to detect user engagement from mouse tracking data. From the analysis and exploration conducted it cannot be fully concluded that the methods presented here can fulfil this aim. The initial results are promising, but without a properly labelled dataset there is no way of confirming these results.

To summarise this project users mouse data from 2 groups of users performing a simple task was analysed and modelled. The different groups conducted the experiment in different ways. 13 of the users conducted the experiment in a lab, where they were closely monitored to ensure they were paying attention. 380 of the users were crowdsourced online using Amazon's Mechanical Turk. These users were not monitored so it is uncertain if they were paying attention. A variety of techniques was attempted, and the use of Hidden Markov Models was evaluated to perform best at this task. Models were trained for each group of users on different sources of data. One a sequence of html element targets and the other a multivariate time series of a users mouse cursor location. Each users data was evaluated on the lab model and the crowdsourced turk model. Some of the crowdsourced users was evaluated to belong more closely to some of the lab study users rather than the other crowdsourced users. This means that these users were doing something different from their peers, they are suspected to be paying attention more than their peers. From the results of the model I can predict that anywhere from 200 to 12 of the crowdsourced users were paying attention, with varying degrees of confidence.

However I would not be too confident in my findings. The data was incorrectly labelled for the task I wanted to perform. Additionally the data was heavily imbalanced. All of the conclusions I have made from the lab data are

relying on only a small number of datapoints which is not statistically significant. There were not meaningful amounts of data from the other class of online data either. Typically, data science and machine learning uses big data, where as this project used only 388 records in total which came to only 330 MB of data in its original JSON format. Any bias in any of these original samples will be magnified as this was used to classify more points, which would again spread this bias. This would become a self-fulfilling prophecy as more similar points would be labelled and spread the belief.

Any concern that the models have learned of a simple feature such as sequence event has been disproven, showing the reclassification algorithm has nonesuch simple relationship.

Biggest flaw of the project is the first assumption, that the lab people were all paying attention. While they were monitored and we can be sure they were not distracted by phones or televisions, many of them may have 'zoned out' and not have been given it their full effort and attention.

References

[1] Amazon. *FAQs About Amazon Mechanical Turk*. 2020. URL: `https://www.mturk.com/worker/help` (visited on 18/08/2020).

[2] Joseph Chee Chang, Saleema Amershi and Ece Kamar. "Revolt: Collaborative crowdsourcing for labeling machine learning datasets". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2017, pp. 2334–2346.

[3] Gabriele Paolacci, Jesse Chandler and Panagiotis G Ipeirotis. "Running experiments on amazon mechanical turk". In: *Judgment and Decision making* 5.5 (2010), pp. 411–419.

[4] Jesse Chandler and Danielle Shapiro. "Conducting clinical research using crowdsourced convenience samples". In: *Annual review of clinical psychology* 12 (2016).

[5] Neil Stewart, Jesse Chandler and Gabriele Paolacci. "Crowdsourcing samples in cognitive science". In: *Trends in cognitive sciences* 21.10 (2017), pp. 736–748.

[6] Panagiotis G Ipeirotis, Foster Provost and Jing Wang. "Quality management on amazon mechanical turk". In: *Proceedings of the ACM SIGKDD workshop on human computation*. 2010, pp. 64–67.

[7] Chien-Ju Ho et al. "Incentivizing high quality crowdwork". In: *Proceedings of the 24th International Conference on World Wide Web*. 2015, pp. 419–429.

[8] Edith Law et al. “Curiosity killed the cat, but makes crowdwork better”. In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 2016, pp. 4098–4110.

[9] Eric T Peterson and Joseph Carrabis. “Measuring the immeasurable: Visitor engagement”. In: *Web Analytics Demystified* 14 (2008), p. 16.

[10] Gruber Julian. “A comparison of user interactions in lab and crowdsourced studies”. University of Vienna, 2017.

[11] Divesh Lala et al. “Detection of social signals for recognizing engagement in human-robot interaction”. In: *arXiv preprint arXiv:1709.10257* (2017).

[12] Andrew T Duchowski. “Eye tracking methodology”. In: *Theory and practice* 328.614 (2007), pp. 2–3.

[13] Georg Buscher, Edward Cutrell and Meredith Ringel Morris. “What do you see when you’re surfing? Using eye tracking to predict salient regions of web pages”. In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2009, pp. 21–30.

[14] Tony Renshaw, Richard Stevens and Paul D Denton. “Towards understanding engagement in games: an eye-tracking study”. In: *On the Horizon* (2009).

[15] Shahram Eivazi and Roman Bednarik. “Predicting problem-solving behavior and performance levels from visual attention data”. In: *Proc. Workshop on Eye Gaze in Intelligent Human Machine Interaction at IUI*. 2011, pp. 9–16.

[16] Daniel Szafir and Bilge Mutlu. “Pay attention! Designing adaptive agents that monitor and improve user engagement”. In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2012, pp. 11–20.

[17] Daniel C Richardson and Michael J Spivey. “Eye tracking: Characteristics and methods”. In: *Encyclopedia of biomaterials and biomedical engineering* 3 (2004), pp. 1028–1042.

[18] Lynne Cooke. “Is the Mouse a” Poor Man’s Eye Tracker”?” In: *Annual Conference-Society for Technical Communication*. Vol. 53. 2006, p. 252.

[19] Urška Demšar and Arzu Çöltekin. “Quantifying gaze and mouse interactions on spatial visual interfaces with a new movement analytics methodology”. In: *PloS one* 12.8 (2017).

[20] Jeremy Goecks and Jude Shavlik. “Learning users’ interests by unobtrusively observing their normal behavior”. In: *Proceedings of the 5th international conference on Intelligent user interfaces*. 2000, pp. 129–132.

[21] Chen-Chung Liu and Chen-Wei Chung. “Detecting mouse movement with repeated visit patterns for retrieving noticed knowledge components on web pages”. In: *IEICE transactions on information and systems* 90.10 (2007), pp. 1687–1696.

[22] David Hauger, Alexandros Paramythis and Stephan Weibelzahl. "Using browser interaction data to determine page reading behavior". In: *International Conference on User Modeling, Adaptation, and Personalization*. Springer. 2011, pp. 147–158.

[23] Vidhya Navalpakkam and Elizabeth Churchill. "Mouse tracking: measuring and predicting users' experience of web-based content". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2012, pp. 2963–2972.

[24] Jeff Huang, Ryen White and Georg Buscher. "User see, user point: gaze and cursor alignment in web search". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2012, pp. 1341–1350.

[25] Matthias Hirth, Tobias Hossfeld and Phuoc Tran-Gia. "Analyzing costs and accuracy of validation mechanisms for crowdsourcing platforms". In: *Mathematical and Computer Modelling* 57.11-12 (2013), pp. 2918–2932.

[26] Alexander J Quinn et al. "Crowdflow: Integrating machine learning with mechanical turk for speed-cost-quality flexibility". In: *Better performance over iterations* (2010).

[27] Carsten Eickhoff and Arjen P de Vries. "Increasing cheat robustness of crowdsourcing tasks". In: *Information retrieval* 16.2 (2013), pp. 121–137.

[28] Xiaojin Zhu and Andrew B Goldberg. "Introduction to semi-supervised learning". In: *Synthesis lectures on artificial intelligence and machine learning* 3.1 (2009), pp. 1–130.

[29] Xiaojin Jerry Zhu. *Semi-supervised learning literature survey*. Tech. rep. University of Wisconsin-Madison Department of Computer Sciences, 2005.

[30] Omid Mohamad Nezami, Debbie Richards and Len Hamey. "Semi-Supervised Detection of Student Engagement." In: *PACIS*. 2017, p. 157.

[31] Daniel Jurafsky and James H Martin. "Speech and language processing (draft)". In: *Chapter A: Hidden Markov Models (Draft of September 11, 2018). Retrieved March* 19 (2018), p. 2019.

[32] Oliver Ibe. *Markov processes for stochastic modeling*. Newnes, 2013.

[33] Michael Collins. "Tagging with Hidden Markov Models". In: (2016).

[34] José Gordillo and Eduardo Conde. "An HMM for detecting spam mail". In: *Expert systems with applications* 33.3 (2007), pp. 667–682.

[35] Lawrence R Rabiner. "A tutorial on hidden Markov models and selected applications in speech recognition". In: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286.

[36] Jaime Ferrando Huertas. *Generating synthetic data through Hidden Markov Models*. 2018.

[37] Charisma Choudhury et al. *State dependence in lane changing models*. 2007.

[38] Jeremy Ronk. "Structured, semi structured and unstructured data". In: *Retrieved November* 21 (2014), p. 2015.

[39] Data Robot Wiki. *Feature Variables*. 2020. URL: https://www.datarobot.com/wiki/feature/ (visited on 16/09/2020).

[40] Zhengzheng Xing, Jian Pei and Eamonn Keogh. "A brief survey on sequence classification". In: *ACM Sigkdd Explorations Newsletter* 12.1 (2010), pp. 40–48.

[41] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[42] Alex Krizhevsky, Ilya Sutskever and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[43] Nitish Srivastava et al. "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[44] Charu C Aggarwal and ChengXiang Zhai. "A survey of text classification algorithms". In: *Mining text data*. Springer, 2012, pp. 163–222.

[45] Utkarsh Porwal, Zhixin Shi and Srirangaraj Setlur. "Machine learning in handwritten Arabic text recognition". In: *Handbook of Statistics*. Vol. 31. Elsevier, 2013, pp. 443–469.

[46] Hmmlearn developers. *hmmlearn*. https://github.com/hmmlearn/hmmlearn. 2020.

[47] Jennifer Pohle et al. "Selecting the number of states in hidden Markov models: pragmatic solutions illustrated using animal movement". In: *Journal of Agricultural, Biological and Environmental Statistics* 22.3 (2017), pp. 270–293.

[48] William S Noble. "What is a support vector machine?" In: *Nature biotechnology* 24.12 (2006), pp. 1565–1567.

[49] Charu C Aggarwal and ChengXiang Zhai. "An introduction to text mining". In: *Mining text data*. Springer, 2012, pp. 1–10.

[50] Beth Sagar-Fenton and Lizzy McNeil. *How many words do you need to speak a language?* 2018. URL: https://www.bbc.co.uk/news/world-44569277 (visited on 12/09/2020).

[51] Dan Jurafsky and C Manning. "Text Classification and Naive Bayes". In: *Vorlesungsskript http://www.stanford.edu/class/cs124/lec/naivebayes.pdf* (2015).

[52] James Martin Daniel Jurafsky. *Speech and Language Processing*. 2009.

[53] Nitesh V Chawla et al. "SMOTE: synthetic minority over-sampling technique". In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[54] Diana Mindrila and Phoebe Balentyne. “Scatterplots and correlation”. In: *Retrieved from* (2017).

[55] Human Computation. *Workers Browser Activity in CrowdFlower Tasks*. 2017. URL: `https://www.kaggle.com/humancomp/worker-activity-crowdflower` (visited on 03/05/2020).