Deception detection using keystroke dynamics

On the methods to predict deceptive behavior by looking at the keystroke rhythm

A.B. Huisman

12 December 2016

UNIVERSITEIT TWENTE.

DECEPTION DETECTION USING KEYSTROKE DYNAMICS: ON THE METHODS TO PREDICT DECEPTIVE BEHAVIOR BY LOOKING AT THE KEYSTROKE RHYTHM

BY

ALBERT BOAZ HUISMAN

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Industrial Engineering & Management
at Universiteit of Twente in Enschede.

Zwolle, The Netherlands

Adviser:

Prof. Dr. Marianne Junger

Dr. Chintan Amrit

Dr. Soumik Mondal

PwC adviser:

J. Aussems

Foreword

The Danish philosopher Søren Kierkegaard once wrote

"*Life can only be understood backwards; but it must be lived forwards.*"

After two years of studying at the University of Twente, I now understand that these years have been the most influential part of my life yet. After a lot of hard work, long nights, many collaborative assignments, and a lot of insights, this phase is coming to an end. During my time at the University of Twente, I have learned more than I could have ever imagined. I discovered new passions like programming and mathematics and I got familiar with academic research. I also discovered that if you cannot think of a solution to a problem right away, this does not mean that you cannot solve the problem at all. In the words of the mathematician Alexander Grothendieck,

"(mathematical) problems are of two sorts: some are like nuts one cracks open with a sudden hard blow; others are like walnuts that one soaks in water for days until the tough skin peals away of itself."

This turned out to be the one of my most important life lessons yet. By taking the risk and uncertainty of working on something that is not immediately obvious or apparent, the most elegant and beautiful solutions reveal themselves to you. With this in mind, I looked for opportunities in Twente to learn and I tried to follow my interests. During the master Industrial Engineering & Management, I took courses from Mechanical Engineering, Computer Science and Applied Mathematics not knowing beforehand if I would be able to finish those courses. And indeed, sometimes I did not finish some courses due to a gap in my knowledge, sometimes I got high grades, but in every case I made sure that I learned something from these courses. Looking back, I have a rich collection of experiences that helped me in finding my strengths, weaknesses, and passions. I am sure these experiences will help me in my further career. I am very grateful that I have had the opportunity to study at the University of Twente.

I would like to thank Prof. dr. Marianne Junger for accepting my initial thesis proposal, helping me find the right subject when the first subject did not appear feasible and guiding me through the process of doing research with her extensive research experience. I am really grateful for all the help I received, the collaborative moments and the quick communication. It was truly a pleasurable experience. Next, I would like to thank Dr. Chintan Amrit who took the effort to sit down and think with me about the research and most of all in assuring me if I was on track and what I was to expect. This was really important to me and helped me to stay confident in the process of writing this thesis. I would also like to thank Dr. Soumik Mondal, who is an expert in Keystroke Dynamics and took the time to sit down with me to share his experience on how to approach the data. From PricewaterhouseCoopers I would like to thank my supervisor Jos Aussems, for thinking with me, helping me with managing this thesis and for asking the right questions to guide me in the right direction. I look forward to working with you in the future.

On a personal level, I would like to thank my significant other, Marjolein Kouwen, who supported and stood by me in good times but also during stressful moments while writing this document. Your resilient personality is inspiring. I would also like to thank my parents, Bert and Erica Huisman, who have supported me during my studies. You have always wished the best for me and I am grateful to have you as my parents.

I hope you will enjoy reading this thesis.

Albert Boaz Huisman

Abstract

This thesis addresses the possibility of using keystroke dynamics to detect deceptive messages without looking at the contents of the message. Keystroke dynamics (KD) is the detailed timing information that describes exactly when each key was pressed and when it was released as a person is typing at a computer keyboard. KD is considered a behavioral biometric. Deceivers often exhibit behavioral and physical traits as a consequence of their deception. In this thesis, it is tested if deceiving causes changes in a deceivers typing rhythm. One recent paper (Banerjee, Feng, Kang, & Choi, 2015) already confirmed this hypothesis with high accuracies, by also considering the content of the message. However, doing so is highly privacy invasive. Therefore, it is useful to analyze if KD solely can be used to detect deception.

First, the literature on deception detection and keystroke dynamics is studied to gain insights in the two research fields. Based on insights from the literature reviews, an experiment to gather data (n = 30) within PricewaterhouseCoopers (PwC) is designed, the characteristics (features) of the data are extracted and methods with which this data can be analyzed are selected and used. The messages will be modeled differently than in the study of Banerjee et al., which does not take into account that keystroke dynamics is a biometric property and consequently is different from person to person. The features that are used in this thesis are dwell time, four flight time variants and the pauses between words. A best-of-three selection method is used to select the three most appropriate features for each participant individually. The (machine learning) methods used are scaled Manhattan distance based metric, Naive Bayes, Support Vector Machine, k-Nearest Neighbor, C4.5, and Random Forest. The corpus of Banerjee et al. is available and is used for comparison to the PwC dataset using the same features and methods. A deviation from Banerjee et al. is that the PwC dataset contains four messages per participant (two truthful and two deceptive) whereas Banerjee et al. only gathered two messages per participant (one truthful and one deceptive).

The best performing algorithm was k-Nearest Neighbor which could successfully tell deceptive and truthful message apart for 13% – 15% of the participants. In most cases for 80% of the participants or more it was not possible to discriminate truth from deceptiveness by the keystroke dynamics of the messages alone. The classification showed an extreme classification bias which means that both messages were either classified as deceptive or truthful. A random sample (n = 100) of the Banerjee et al. corpus seems to confirm this finding as the accuracies are almost identical. To conclude, it did not seem possible for most participant using these datasets, features and methods to discriminate between truthful and deceptive messages.

Table of Contents

List of figures

List of tables

List of acronyms

ANN	Artificial Neural Network
CART	Classification And Regression Tree
CMC	Computer-Mediated Communication
DD	Deception Detection
DOM	Document Object Model
HCI	Human-Computer Interaction
KD	Keystroke Dynamics
MD	Mouse Dynamics
ML	Machine Learning
NB	Naive Bayes
RF	Random Forest
SVM	Support Vector Machines
VSA	Voice Stress Analysis
PwC	PricewaterhouseCoopers

Glossary

Confusion matrix	A matrix consisting out of four classification categories where the total number of each category is presented.
Document Object Model	An object orientated approach of structured elements, e.g. HTML.
Dwell Time	The exact key press duration.
Feature	A characteristic of the data (e.g. typing speed)
Four-Factor Theory	An elaboration on the leakage hypothesis that describes the variables that accumulate the leakage hypothesis. These variables are arousal, negative affect, cognitive effect and behavioral control
Flight time	The time between the press- and/or release combinations of two (or more) keys.
Instance	A data point.
JavaScript	A client-side programming language for the browser.
jQuery	A JavaScript library that contains a lot of JavaScript functionality.
Key event	A keypress resulting in a consecutive keydown and keyup event of a certain key.
Key press	The event where the computer registers that a key is pressed.
Key up	The event where the computer registers that a key is released.
Leakage Hypothesis	A hypothesis that states that liars would experience involuntary physiological reactions driven by increased arousal, negative affect, and discomfort that would “leak out” in their nonverbal behavior cues.
Milgram Experiment	A study done by psychologist Stanley Milgram to measure the willingness of participants to obey the instructions of an authority to perform acts conflicting with their personal conscience.
True/False Positive/Negative	Correct (true) or false (false) classification of an instance (data point) to either the positive or negative class.
XMLHttpRequest	A request initiated from the client side using JavaScript to make a HTTP request to another page.

Why do almost all people tell the truth in ordinary everyday life? Certainly not because a god has forbidden them to lie. The reason is, firstly because it is easier; for lying demands invention, dissimulation, and a good memory.

Friedrich Nietzsche, *Human, All Too Human*, II.54, 1878/1996

1. Introduction

1.1. Background

Deception is defined as "a message knowingly transmitted by a sender to foster a false belief or conclusion by the receiver" (Buller & Burgoon, 1996). Using this definition, deception may take a variety of forms ranging from pure fabrication to half-truths, vagueness and concealments (Carlson, George, Burgoon, Adkins, & White, 2004). Over the course of centuries, humans have been trying to read between the lines and crack the code of deception. Deception is ubiquitous and is often used to gain an advantage over others. The scientific field of deception detection is built upon hypotheses and theories, for example the leakage hypothesis (Ekman & Friesen, 1969) and the four factor theory (Zuckerman, DePaulo, & Rosenthal, 1981). Conclusive statements are difficult to make because many theories are connected to human traits (e.g. emotions) that are not fully understood yet. Thorough research is also difficult because it is hard to encounter real life situations where genuine deception can actively be monitored. Many studies are focusing on physiological- and behavioral changes because these traits are observable and measurable. For example, it was found that deception can be recognized by looking at the dilation of the pupils (Wang et al., 2010) and by monitoring the pitch of the voice (Patil, Nayak, & Saxena, 2013).

Since the rise of the Internet, deception has found its way into computer-mediated communication (CMC). A lot of people have fallen prone to malicious digital actors through email, chat sessions or other applications. The anonymity the internet provides has caused a lot of misdemeanour. Scammers can send fake emails to persuade vulnerable receivers to enter their credentials. The insurance industry suffers from false claimants, who can now submit a claim through a website or online form. Some users in chatrooms take on different identities to prey on inexperienced users which sometimes escalates to harmful events like extortion. In a lot of cases, deception is used to intentionally send a message that fosters a false belief by the receiver. The difficulty in recognizing a deceiver in a digital environment is that, aside from the written text, there are no clues that can indicate deception. In real life communication, blushing or gaze aversion is often perceived as a clue to indicating deceptive intent (Vrij, 2008). In a CMC environment, these traits are non-apparent and the receiver's only option is to classify the intent of a message based on its content. Since the keyboard and mouse are some of the few (or only) input devices that users on the Internet have, it would be useful to assess whether these input devices can yield clues that can help in assessing the intent of a message. If the intentions of a deceiver can be determined on forehand by clues from the keyboard or mouse, then users can be protected from self-inflicted damage by acting upon malicious intent.

Keystroke dynamics is the detailed timing information that describes exactly when each key was pressed and when it was released as a person is typing at a computer keyboard.[1] Keystroke dynamics has proven to be rather useful as a biometric in research to authenticate or even identify unique users. Behavioral biometrics often have the advantage of being unobtrusive but are considered far more fallible than physiological biometrics (Revett, Gorunescu, Gorunescu, Ene, & Santos, 2007). Keystroke dynamics does also not meet the European access control standards such as EN-50133-1 (Rybnik, Panasiuk, Saeed, & Rogowski, 2012) yet which makes the application of the behavioral biometrics not suitable for first-step verifications. The technique is often combined with other forms of authentication, for example second step authentication where not only the correct password is necessary but also the right typing metric. The research on keystroke dynamics has a strong focus on authentication. About 89% of the papers focus on authentication, where 5% focus on identification and in 6% of the cases it is not explicitly mentioned (Teh, Teoh, & Yue, 2013). But the scientific community has also turned towards more applications than just identification and authentication, such as emotion recognition (Epp, Lippold, & Mandryk, 2011; Vizer, Zhou, & Sears, 2009). These applications could yield valuable insights with regard to online marketing.

As mentioned earlier, often behavioural changes indicate deception. There is a need for deception detection in CMC environments as the Internet. Considering that users often only use a mouse and keyboard in these environments, it would be useful to study the relation between typing behaviour (as described by Keystroke Dynamics) and deception. It can therefore be hypothesized that the typing behaviour (KD) of an individual that writes a deceptive message differs from the typing behaviour when he writes a truthful message. Looking at deception detection, there are many examples of (sometimes unexpected) behavioural changes when deceiving, like an increase in pause duration or a decrease in response length (Vrij, 2008). Such characteristics could also be apparent in typing behaviour.

[1] https://en.wikipedia.org/wiki/Keystroke_dynamics

At the time of writing, only one paper is published that used high level keystroke dynamics to help classify deception (Banerjee et al., 2015) in addition to another approach (stylometry). The results of this study were quite high, with classification accuracies of over 90%. However, a common semantical deception detection technique (that looks at the word usage) as a baseline raised the accuracy up to 80% higher. The quantitative keystroke dynamics features (like message length and deletion key usage) therefore only increased the accuracy with a few percent. While looking at the contents of a message is privacy invasive, it may be more interesting to see how accurately only KD can be used to perform a more in depth analysis can help in understanding how deception and typing behaviour are related. For PricewaterhouseCoopers (PwC), the relevance of this study lies in the business value. If deceptive behavior can be assessed by looking at the keystrokes, then PwC can turn this technique into a business case. Using keystroke dynamics for deception detection could yield an interesting value for assessing the validity of online reviews or for insurance companies who want to be able to automatically assess the validity of claims. Insurance companies could greatly benefit from distinguishing deceptions as fraudulent claims cost billions of euros annually.

1.2. Objectives

The objective of this thesis is assess if deception can be detected using keystroke. This can be done by studying the relevant literature and to test the gained insights in practice.

By understanding these two fields of study and the conjunction between them, the opportunities of correlating the two can be further explored. In order to succeed in finding an appropriate approach of correlating the two fields of study, a relevant in-depth literature review is necessary. These in-depth studies yield a lot of knowledge on both subjects. Studying the theory of deception and the challenges of doing research in this domain yields insights in what can be expected of deception and how people respond to it. A description of the state of Keystroke Dynamics as well as successes in the applications and the common approaches yields insights in what characteristics can be derived from the data and what methods can be used to analyze this data.

Once a theoretical fundament is established, the theory will be tested out in practice. It is known that deception induces changes in behavior and/or physiological traits. Sometimes these changes are so subtle that they cannot be easily detected by a human. Keystroke dynamics may be such an indicator. Changes in the typing behavior, consisting out of multiple keystrokes per second, cannot be processed easily by a human brain. Therefore, computer supported analysis is done to analyze all the keystrokes.

Once the data is collected and the data is analyzed, the results can be compared to the dataset (corpus from (Banerjee et al., 2015)). The researchers already reported high accuracies using stylometry. This thesis will assess whether discrimination between truthful and deceptive messages is also possible without looking at the contents of the message, i.e. without using stylometry.

1.3. Approach

This paragraph will discuss what methodology is used and how the objective can be achieved.

According to a popular poll conducted by KDNuggets[2], there are a few methodologies available for data science projects. The poll reveals that most studies (43%) follow the CRISP-DM methodology for their data-mining projects. However, there is an increasing trend of researchers who use their own methodology (28%). CRISP-DM is a very useful method to give structure to a data-mining project. CRISP-DM assumes there are data available and a research question as driving force. Also, the refined and extended successor ASUM-DM[3] retains the method but focuses also on the infrastructure/operations side of implementing a DM project. However, the method on its own does not indicate how to set up a research design to collect data. Therefore, the approach in this thesis will be influenced but not guided by the CRISP-DM methodology, and thus follows a self-defined approach. This approach starts with a literature review, data collection, followed by data preparations, modeling, and evaluation. Deployment is not a goal of this thesis, as it is uncertain whether patterns will emerge and there is only a limited amount of time. This approach has more similarities with the approach of (Shmueli & Koppius, 2011). This is done following the sequential steps of goal definition, data collection & study design, data preparation, exploratory data analysis, choice of variables, choice of methods, evaluation,

[2] http://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data-mining-data-science-projects.html

validation, model selection and finally model use and reporting. Aside from a change in the order, this method will decide the structure of this document.

First, a literature study is done to gain knowledge about the subjects Keystroke Dynamics and Deception Detection. This literature overview should give a good impression on the research and the current state of both fields. Afterwards, the research questions can be formulated. When the research questions are formulated, a facilitating research will be designed.

This research design will be key to generating a dataset. The executive part of the research that is designed will be exposed to PwC employees. The research has to comply with certain conditions resulting from the literature study.

After enough participants have completed the research, the data can be modeled and explored to review the characteristics. Preparing and modeling the data is then necessary because the raw data from the webserver needs to be processed into a useful format, from which the characteristics can be easily extracted. Also, the data should be modeled meaningfully with an eye on the methods that will be used. The literature on Keystroke Dynamics describes ways to extract useful characteristics from the data. These characteristics will be explored in the context of deception detection.

The exploratory analysis should indicate which characteristics are useful for the classification of deception. These characteristics can then be used with the methods that come forth out of the KD literature.

1.4. Scope

Since there is an overlap of two fields, data science and psychology, a clear scope is important to formulate answers to the research question. First, in this thesis the focus will not be on hypothesizing new theories about deception. The literature on deception is used as necessary background and to find a way to design an experiment to generate data. Philosophical discussions about deception will therefore not be handled in this thesis. This thesis may be considered mainly a data science project, which means that the focus is on data science and finding patterns in the data. Therefore, the current knowledge about keystroke dynamics will be thoroughly studied and possible new insights may be generated during this thesis. Mainly, knowledge of both fields will be applied for this application. This thesis is written with the assumption that KD is a biometric (Moskovitch et al., 2009). The goal of this thesis is to design an experiment, test the data and answer the research question.

Looking at the keystrokes of a user is a highly privacy sensitive subject. As keystroke dynamics may be considered a biometric, logging keystroke behaviour may be equivalent to logging biometrical data of individuals. This thesis will not deal with the privacy consequences that correlating deception and keystroke dynamics may imply. This thesis will also not deal with the contents of the message, as it tries to look for patterns without using the semantical meaning of the message.

1.5. Thesis structure

In chapter 2, a literature study is done on keystroke dynamics and deception detection to generate ideas for the experiment and to assess the current state of both research fields including the hypothesis that are established. The methods with which the data can be analyzed are also discussed. In chapter 3, the research question is formulated and a research design is described. A data collection method is explained that is used to create a dataset based on the results of chapter 2. In chapter 4, the appropriate keystroke characteristics from literature are treated and selected. In chapter 5, the data is explored and possible relevant features are tested to find patterns that may hint on deception. In chapter 6, the data analysis phase is explained. In this chapter the data is analyzed using the selected methods and the first results are presented. In chapter 7, the results are analyzed and discussed. In chapter 8, the research question is answered and ideas for future work are outlined.

[3] https://developer.ibm.com/predictiveanalytics/2015/10/16/have-you-seen-asum-dm/

2. Review of Related Literature

This thesis lies in the intersection of two topics: keystroke dynamics and deception detection. In sub-chapter 2.1 and 2.2, the literature of Deception Detection (DD) and Keystroke dynamics (KD) will be studied respectively. In sub-chapter 2.3, successful methods with which keystroke data is analyzed will be explained and selected for further use. Then in sub-chapter 2.4, the performance measure with which the results of the methods can be analyzed will be explained.

2.1. Deception detection

2.1.1. Scope

The concept of deception can be defined as "a message knowingly transmitted by a sender to foster a false belief or conclusion by the receiver" (Buller & Burgoon, 1996). When this definition is used, deception can take a variety of forms ranging from pure fabricated lies to half-truths, vagueness and concealments (Carlson et al., 2004). Deception detection has proven to be a difficult terrain to study. Over the last decades a lot of researchers have looked for ways to distinguish truth from lies.

Deceptive communication can be detected by considering different categories of cues. There exist verbal cues (e.g. language style or message content), nonverbal cues, contextual cues and meta-cues (Carlson et al., 2004). Verbal and contextual cues will not be considered in this thesis. Meta-cues are typically detectable interaction between two or more of sets of cues that itself will serve as an additional cue. Since deceivers are in charge of their behavior, they may strategically adapt it to mask their deceit. Ambiguous change in multiple cues may indicate deception. However, meta-cues are not considered in this thesis because the feature is too advanced for the analysis that will be done in this thesis. The scope of this literature is to look at the nonverbal behavior, specifically keystroke dynamics, to estimate if it contains cues that might indicate deceptive behavior.

2.1.2. Theoretical approach

To find why measurable differences between a truth teller and a deceiver occur, it is useful to understand the theoretical framework that researchers have established. The theoretical framework describes the causal variables that accumulate behavioral changes. In 1969, it was hypothesized that liars would experience involuntary physiological reactions driven by increased arousal, negative affect, and discomfort that would "leak out" in their nonverbal behavior cues (Ekman & Friesen, 1969; Elkins, Zalfeiriou, Burgoon, & Pantic, 2014). The leakage cues reveal what liars are trying to hide, for example how they really feel. Whereas the deception cues indicate if deception is occurring, without spoiling the type of information that is being concealed. Building upon this hypothesis was the four-factor theory (Zuckerman et al., 1981) which postulated four potential causes of leakage: Arousal, Negative affect, Cognitive effect and Behavioral control. It is important to state that this model is limited to behavior that can be discerned by human perceivers without the aid of any special equipment (DePaulo et al., 2003).

From these four factors, arousal has the most dominant role. It is theorized that a person who engages in deceit finds that to be distressing. This results in an increased level arousal. The relation between deception and arousal however is not deterministic. Deceit does not inevitably trigger arousal. There are many lies that perpetrate everyday life, like giving compliments for the benefit of others, which do not evoke arousal. Arousal is not always detectable, as people are able to mask their inner feelings to a certain degree. Another important aspect is that other factors can also cause arousal. For example, a person can experience arousal by telling a difficult truth which may cause behavior that is also present when being deceptive (e.g. increase in pauses). Lastly, behavior during arousal may vary from person to person (Elkins et al., 2014).

Negative affect means that the deceiver generally has a feeling of guilt or fear when deceiving. Cognitive effect stems from the prediction that lying is a more cognitive complex task than telling the truth, a cognitive burden the deceiver can be aware of. (DePaulo et al., 2003). Lastly, deceivers may also try and control their behavior in such a manner that it becomes unnatural. These mechanisms have been richly studied whereas researchers have mostly focused on manifestations of these mechanisms. It was Ekman (1985 – 1992) who conceptualized the role of emotions in deceiving. He stated that by understanding the emotions that liars feel, it may be possible to predict behavior that may distinguish liars from truth tellers. Think of guilt and fear when a deceiver lies, as a driving force for changes in behavior (e.g. speech or muscular activity).

2.1.3. *Human evaluation*

The most common way to evaluate the performance of deception detection is done by placing a person in front of a group of peers and instruct the person tell a lie. In one of the earlier papers on deception detection, 32 persons answered four questions in front of six peers with randomly assigned high or low motivational conditions. The difference in motivational conditions for deception is due to the fact that many of the lies perpetrated in daily life are uninvolving nor arousing. The research showed that lies with highly motivational conditions were harder to detect verbally, but more readily detected when non-verbal detectors were available. Lies that were planned on forehand were no more or less detected than lies that were not planned. Planned responses however, were perceived as more deceptive, more tense and less spontaneous by the judges (DePaulo, Lanier, & Davis, 1983). This study indicated a change in behavior when a person is motivated to lie and this behavior often exhibits sub-conscious changes in behavior. The accuracy to distinguish deception from truth is often compared to the probability of guessing, with a measured average of 54% as research has shown (Bond & DePaulo, 2006). It is studied however, that professionals in lie detection are much more accurate in detecting a lie then the average layperson when behavioral clues can be detected in real time (Ekman, O'Sullivan, & Frank, 1999). Deception detection has been studies in forensic contexts, but researchers have found that other areas are equally relevant. For example, deception detection has been studied at an insurance company. It has been showed that operators were only able to correctly classify 50% of the false claimants over the telephone. In the study, claimants said little and both truthful and deceptive statements were equal in quality based on the Criteria-Based Content Analysis (CBCA) (Leal, Vrij, Warmelink, Vernham, & Fisher, 2013). Another study showed that there is an improvement of deception detection when people get trained to detect lies. Training makes a difference in lie detection performance. It did not seem to make a difference if the person is trained by electronic means or by traditional lecture-based delivery. The results are the same (George et al., 2004). In another study, a specific experiment (i.e. Concealed Information Test) was evaluated to be useful to detect criminal intent. It can be concluded that humans are bad performers in the detection of deception.

2.1.4. *Other methods*

In order to enhance the effectiveness of deception detection, researchers have turned to other tools to discern lies from truth. The best known method is the polygraph which detects changes in autonomic reactions by measuring bodily functions like respiration rate, skin conductivity, heart rate, blood pressure, capillary dilation and muscular movement.[4] The tool was primarily developed between 1895 and 1945 and is still the most used method. The autonomic reactions are hard to control by the conscious mind and can give away deception. Because the protocol for administering the polygraph examination requires a lengthy (3 – 5 hours), multiphase interview to obtain reliability, and because background investigations are often preceded, the polygraph is unsuitable for rapid screening environments and automation (Elkins et al., 2014). The evaluation of the results of the polygraph are often performed manually. The polygraph does give an indication on the scoring and the probability of deception, however most examiners base their decision on their own judgement of the scores. When considering laboratory studies, it was suggested that the polygraph tests is about 82% accurate at identifying deceit. In 16% of the cases a deceiver would be falsely indicated as innocent. From the innocent group 88% was correctly classified. The false positive rate of falsely accusing an innocent participant was 9% (Vrij, 2008). However, often those laboratory results are overestimations as the experiments are too sterile. The real accuracy is often much lower. More recent and proving to be more effective, is the Voice Stress Analysis (VSA). Using this method, stress can be inferred by speech. It is shown that VSA performs better than the Polygraph in the detection of stress (Patil et al., 2013). Stress does not automatically infer lying, but a 18-year long field study has shown that stress has a strong predictive force when it comes to deception. A random sample of 279 subjects consisting out of suspects, criminals, defendants, persons of interest and court-ordered mandates were interviewed along with a VSA. The results revealed that a population was tested where 91.7% of the participants were deceptive. Of those tested who were deceptive, 100% had a stress indication. Also, all of the subjects where no stress was indicated by the VSA, were later exonerated from any wrongdoing. In 95% of the cases, VSA could correctly predict the true intent of a subject (Chapman, 2012). VSA is now also considered as an important decision support tool to make a sophisticated estimation of deceptive intent. VSA is being applied to call centers of insurance companies to indicate the validity of a claim. There are more examples of behavioral metrics that have been studied. There is linguistics, where researchers have developed an automatic linguistic tool that analyses text and searches for deceptive clues. This technique looks at the words of the deceptive message (i.e. to assess what a person is saying). Another way to discover deceivers is by looking at the eye behavior, blinks, body posture, gesture and movements. Facial expressions are also a large terrain of study.

[4] https://en.wikipedia.org/wiki/Lie_detection

Since no tool seems reliable enough to conclude a false testimony, most tools are used supportive to final human judgement.

2.1.5. Deception detection in Computer-Mediated Communication

Since the rise of the internet, the popularity of Computer-Mediated Communication (CMC) has expanded voraciously. In a lot of cases, CMC is even the preferred way of communication over real-life communication. The anonymity the internet provides creates the perfect breeding place for deception. Email, chat, and online forms are just some examples of the many possibilities of Computer-Mediated Communication over the internet. These are also examples of where deception takes place. Deception mediated by the computer takes a whole different form then real-life deception.

There has been attention for research on deception in CMC. Nonverbal cues such as vocal pitch, gestures or facial expressions are often not included in this type of communication. As stated earlier, CMC comes with a different context than real-life communication. Nonetheless, research shows that people perform just as bad (or worse) to detect deception on the computer as they do in real life. One study showed that 60.3% of the test group (n=93) failed to detect a fake web shop. Out of this group 30 missed the deception where 26 issued a false alarm (Grazioli & Wang, 2001). In later study by the same researcher, the same concept was applied to a group of MBA students (n=80). Using a one-way ANCOVA the researchers were able to prove that the subjects could not discriminate between the clean and the deceptive site (Grazioli, 2004). Most studies were focused on the reasons why people fall for deception, like fake web shops and phishing campaigns.

2.2. Keystroke dynamics

2.2.1. Chronology

The origins of keystroke dynamics came from the time when telegraphs were introduced. Every sender exhibited a certain rhythm, or signature, by which the experienced receiver could recognize the sender. The same way an autograph can be distinguished uniquely to assert endorsement while the authority may not be physically present. This biometric migrated from and to other forms of communication until the first statistical research was done in 1980 (Gaines, Lisowski, Press, & Shapiro, 1980). The experiments were conducted on seven secretaries in which they were asked to retype the same three paragraphs at two different times over a period of four months. The results were promising but the sample size was too small for a significant statistical result (Monrose & Rubin, 2000). The research ignited the curiosity of researchers because the publications started rising the next years, as shown in Figure 1.

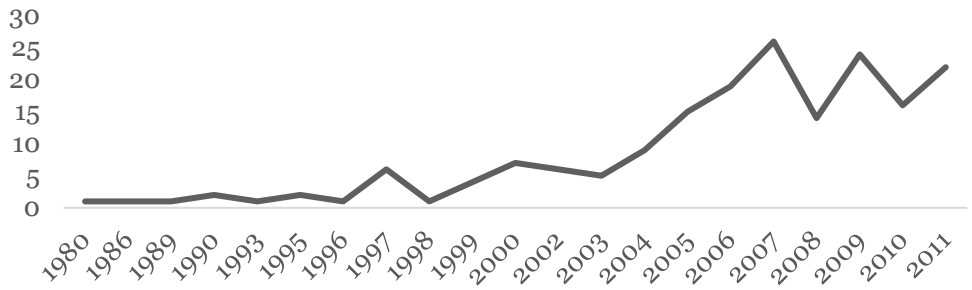

Figure 1 - Number of publications per year on keystroke dynamics (Teh et al., 2013)

2.2.2. Authentication and identification

A survey of 187 papers will be used to describe the current state of KD in the scientific community in this sub-chapter (Teh et al., 2013). This survey gives insight on how researchers have set up their experiments. When it comes to device freedom, 35% reported the usage of a predefined standard device against 17% where the user's own device was used. In terms of platform usage, 44% of the experiments was done by logging from the OS where 17% was done via the web. From all the experiments about 83% performed static keylogging where only 10% were continuously logging. Also, about 33% of the experiments had a number of participants of 20 or less, about 50% was between 21 and 50 participants. However, sample collection can be divided into several sessions over a period of time. This reduces the initial load for the participant but also reflects typing variability.

According to the survey, there are many methods with which the KD data is analyzed. Methods that have been used for KD vary from the most popular distance based metrics like Euclidean (Giot, El-abed, & Rosenberger, 2009), Manhattan and Mahalanobis distances to other statistical methods like the (weighted) probability measures (Monrose & Rubin, 1997), k-Nearest Neighbor, Bayesian (Monrose & Rubin, 2000), Hidden Markov Model (Gould, 2005) and Gaussian Density Function (Lau, Liu, Xiao, & Yu, 2004). Machine Learning techniques were also a popular candidate. Common techniques as the neural network (Revett et al., 2007) showed great success in authentication. Other methods often employed were decision trees, fuzzy logic (Mondal & Bours, 2014), Support Vector Machine (Xiaojun, Zicheng, Yiguo, & Jinqiao, 2013). Statistical methods accounted for approximately 61% of the studies where machine learning was used in about 37% of the studies. Generally, the classification accuracies are quite high whereas some studies achieve an accuracy over 95%.

The sizes of the datasets that have been gathered for analysis in most cases the number of participants was either smaller than 10 (31%) or the number of participants was simply not known (30%). In another 20% of the cases, the number of participants was a number between 11 – 20. There are some datasets freely available either as a benchmark (Killourhy & Maxion, 2009) or to test different algorithms e.g. GreyC dataset (Giot et al., 2009). The dataset that was used to detect deception has also been released (Banerjee et al., 2015) and is freely available.

Over the years, authentication and identification using keystroke dynamics has proven to be very successful. It is therefore a logical consequence to see the technique applied in the industry. Next to many startups trying to exploit the technique, larger companies have also embraced KD.

2.3. Methods

The literature shows that there have been a lot of different approaches to analyze the keystroke data ranging from classical statistical methods to advanced machine learning approaches. Machine learning methods have the advantages of finding relations in complex data. In this sub-chapter, the methods that will be used in this thesis are discussed.

2.3.1. Distance metric

In order to perform the classification, a distance metric can be used. Distance metrics are often used for keystroke dynamics, the most common being the Euclidean, Manhattan and Mahalanobis metric. As the Mahalanobis takes the covariance into account, it is not suitable for this classification task. According to a benchmark study, the (scaled) Manhattan metric outperforms the (scaled) Euclidean metric (Killourhy & Maxion, 2009). Scaling (or normalizing) is important because some features attain a different value range than others.

In the training phase, the mean m_i and the mean absolute deviation s_i of each feature in the feature set of the training data of a participant is calculated. The scaled Manhattan distance metric d is calculated as

$$d = \sum_{i=1}^{n} \frac{|m_i - y_i|}{s_i}$$

where i denotes the i-th element in the mean vector $\boldsymbol{m}$, the test vector of an instance $\boldsymbol{y}$ and the mean absolute deviation vector $\boldsymbol{s}$. A key event from the test set is then classified as belonging to the set for which the distance to that set is the smallest. For example, if the distance to the deceptive dataset is smaller than the distance to the truthful set, the key event is classified as deceptive. Consider the key event from the test set $\boldsymbol{y}$ that has a Manhattan distance d_d with respect to the deceptive training set and a Manhattan distance d_t with respect to the truthful training set. Then the key event $\boldsymbol{y}$ is classified according to the smallest distance.

$$\boldsymbol{y} = \begin{cases} deceptive & d_d < d_t \\ truthful & d_d > d_t \end{cases}$$

If more key events of a message are classified as deceptive, the whole message can be considered deceptive. However, if more key events of a message are classified as truthful, then it is more probable to consider the messages as truthful.

2.3.2. *Choice of algorithms*

The classification problem of this thesis is called a two-class (or binary) classification problem. For this specific problem, several supervised machine learning algorithms can be used. There exist many algorithms and there is not a conclusive way to find out which algorithm is the best. Choosing the best Machine Learning algorithm is in some aspects almost more a craft than it is a science. Some methods that were used, as described by a recent survey, are k-Nearest Neighbor (kNN), k-Means methods, Bayesian classifiers, fuzzy logic, Boost learning, Rnadom Forests, Support vector machine (SVM), Hidden Markov Methods and Artificial Neural Networks (Zhong & Deng, 2015). A survey studied the 10 most influential algorithms in the research community (Wu et al., 2008). These algorithms were C4.5, k-means, Support Vector Machines (SVM), Apriori, EM, PageRank, AdaBoost, k-Nearest Neighbor (kNN), Naive Bayes (NB) and Classification and Regression Trees (CART). Now, not all these algorithms are suitable for a two-class classification problems.

The algorithms that are suitable are Naïve Bayes, SVM, C4.5 (decision tree), kNN and lastly Random Forest (RF) which falls under the CART umbrella and is most suitable for this task. In this paragraph, these algorithms will be explained.

2.3.3. *Naive Bayes*

Naive Bayes classifiers are a family of supervised learning methods based on Bayes' theorem. This conditional probability model has the naïve assumption that every pair of features is independent. Bayes' theorem states that

$$P(y|x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n|y)}{P(x_1, \dots, x_n)}$$

Given a class y and a dependent feature vector $\boldsymbol{x}$**.** By using the naïve assumption that

$$P(x_i|y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i|y), \qquad \forall i.$$

The relation can then be simplified into

$$P(y|x_1, \dots, x_n) = \frac{P(y)\prod_i^n P(x_i|y)}{P(x_1, \dots, x_n)}$$

Since the divisor on the right side of the equation is constant given the input,

$$P(y|x_1, \dots, x_n) \propto P(y)\prod_i^n P(x_i|y) \Rightarrow \hat{y} = \arg\max_y P(y)\prod_i^n P(x_i|y)$$

This technique can be reformulated in the sentence: given the previous instances and the total chance of being in the class y, what is the highest probability of the feature being in the class.

Naïve Bayes (NB) is a famous approach as the model is easy to understand and functions quite well in practice despite its naïve independence assumption. NB is not the ideal candidate for classifying key strokes as Naïve Bayes was originally designed to handle categorical features. Although not the perfect, NB is a very quick and often effective method.

There are roughly three different NB models: Gaussian, Multinomial and Bernoulli. Since the features mostly contain continuous features (as will be shown later in the thesis), the multinomial and Bernoulli models are not suitable for the classification task as these models work exclusively with nominal and binary features respectively. For that reason, only the Gaussian approach is used.

2.3.4. *Support Vector Machine*

Support Vector Machine is an algorithm based on the Vapnik-Chervonenkis theory about statistical learning. The algorithm is fit for regression and classification and is mostly used for linear problems but can be extended to non-linear problems as well. The algorithm is very popular as it is considered flexible and fast while accurate.

The goal of a linear SVM is the creation of a hyperplane that functions as a decision boundary to make binary classifications for p-dimensional instances. SVM does not only create a hyperplane that is able to classify the training instances, but also searches for the hyperplane with the best fit. The best fit is found by maximizing the distances of the instances between the two classes perpendicular to the hyperplane.

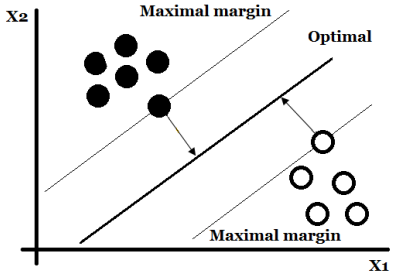

Figure 2 - SVM with a maximal margin hyperplanes and optimal hyperplane

Consider a dataset of n instances

$$(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)$$

where $\boldsymbol{x}_i$ is an p-dimensional vector and $y_i \in \{-1,1\}$ indicates the class of the instance. A hyperplane has to be found that divides the instances based on their class and maximizes the margin between the instances. The margin is the distance to the closest instances $\boldsymbol{x}_i$ for both classes. Since the assumption is that the data is linearly separable, the hyperplane can be described by

$$\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$$

where $\boldsymbol{w}$ is normal to the hyperplane and $\frac{b}{||\boldsymbol{w}||}$ is the perpendicular distance from the hyperplane to the origin. First, two hyperplanes that are closest to the two different classes with no instances in between them can be described by

$$H_1 = \boldsymbol{w} \cdot \boldsymbol{x} + b = 1, \qquad H_2 = \boldsymbol{w} \cdot \boldsymbol{x} + b = -1$$

The distance between the two maximum margin hyperplanes is equal to $\frac{2}{||\boldsymbol{w}||}$. This objective is to minimize $||\boldsymbol{w}||$. No data points should fall between the margins of the hyperplanes creating a constraint for each class. For class $y = 1$, the equation $\boldsymbol{w} \cdot \boldsymbol{x_i} - b \geq 1$ should be satisfied and for class $y = -1$, the equation $\boldsymbol{w} \cdot \boldsymbol{x_i} - b \leq 1$ should be satisfied. Due to the construction of y, this can be written as

$$y_i(\boldsymbol{w} \cdot \boldsymbol{x_i} - b) \geq 1 \; \forall i$$

However, this constraint will not always be satisfied since real data is often not fully linearly separable. Therefore the constraint can be relaxed slightly to allow for misclassified points. This is done by introducing a positive slack variable ξ_i such that

$$y_i(\boldsymbol{w} \cdot \boldsymbol{x_i} - b) - 1 + \xi \geq 0, \qquad where \qquad \xi_i \geq 0, \;\; \forall i$$

Maximizing it subject to the constraints yield the following optimization

$$\min ||\boldsymbol{w}|| + C \sum_i \xi_i \qquad such\ that \qquad y_i(\boldsymbol{x}_i \cdot \boldsymbol{w} + b) - 1 + \xi \geq 0 \;\; \forall i$$

where the parameter C is in charge of the degree in trade-off between the size of the margin and the penalty for the slack variable. The non-linear implementation of the classifier is done by applying the kernel method to the optimal hyperplanes. This means that every dot product is replaced by a nonlinear kernel function. There are many implementations of kernels with the most common being the polynomial and Gaussian radial based function.

2.3.5. *K-Nearest Neighbor*

K-Nearest Neighbor (kNN) is a non-parametric method that can either be used for classification as well as for regression. The general idea is to classify instances by looking at the k nearest neighbors (usually based on the Euclidean distance). If the majority of the neighbors belong to a certain class, then the instance is also classified as such. Often, a weight is assigned to the neighbors to account for the differences in proximity making close neighbors more important than distant ones.

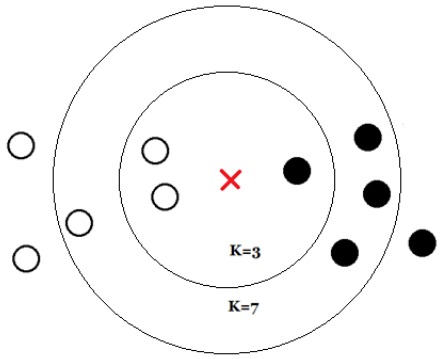

Figure 3 - Example of kNN with k = 3 and k = 7

Consider the example in Figure 3 where no distance weighing is applied. If $k = 3$ is chosen, then the instance at the center of the inner circle will look at the three instances in its vicinity and notice that there are two empty circles and one solid circle. The instance will then be classified as an empty circle. Increasing the k to 7 yields a classification to a solid circle, as there are more solid circles. This becomes very useful if there are random instances in the multi-dimensional features space. The algorithm is fast and effective, making it very popular. Mathematically, the algorithm can be defined as follows. Consider the feature pairs

$$(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)$$

where $\boldsymbol{x}_i$ denotes the vector of features of an instance i whereas y_i indicates the class of said instance either -1 or 1. Given some classified instance $\boldsymbol{x}$ without classification, reorder the known instances such that $||\boldsymbol{x}_1 - \boldsymbol{x}|| \leq \dots \leq ||\boldsymbol{x}_n - \boldsymbol{x}||$. Then the instances can be selected up to k, and with this new reordering, the class can be decided by $y = sgn(\sum_k y_i)$ where the sign function outputs either a -1 when the sum is lower than 0 (and more -1 class members are in the vicinity) and 1 when higher. In order to not classify an instance as class 0, the number k should ideally be uneven. Obviously, the known instances are in the training set and the unknown instances are in the validation set. To account for instances further from the instance itself, weighing is often applied with a factor $\frac{1}{distance}$, making instances further away less relevant to the classification outcome.

2.3.6. *Decision Trees: C4.5 and Random Forest*

Decision trees are suitable for classification and regression jobs. Some well-known and highly effective algorithms are the C4.5 and the Random Forest. CART is often used as an acronym for Decision Tree. CART implementations are very similar to C4.5 whereas the only difference is that CART constructs a tree based on a numerical splitting criterion which is recursively applied to the data. From CART, Random Forest is a well-known and effective algorithm.

Decision trees are flexible and intuitive objects used in classification and regression. The goal of a decision tree is to predict the class (or value) of a target instance based in the features of that instance. The tree consists out of nodes, branches and leaves. A node is a decision rule corresponding with one of the features of the vector. The node then branches to different nodes or leaves depending on the value of a specific feature of an instance. Leaves represent the lowest nodes which do not further branch but assign a conclusive value to the node. If the target variable can attain continuous values then these trees are called regression trees. Other often occurring types are binary trees.

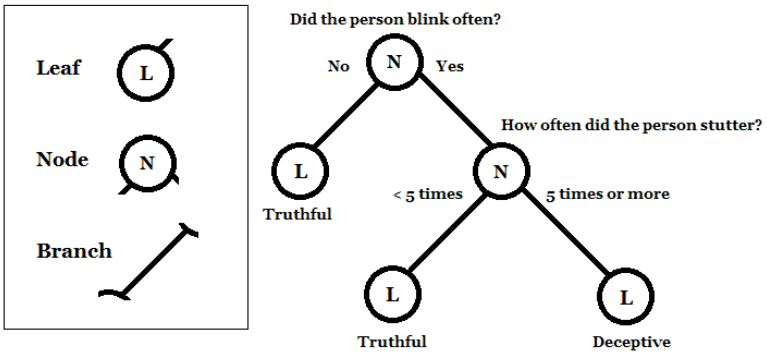

Figure 4 - Simple example of an decision tree with nominal and continuous decision nodes

A decision tree is built from the training set using the concept of entropy and information gain. The tree is constructed top-down from the root node. Consider the training set T consisting of instances t_i

$$T = t_1, t_2, \dots, t_n$$

Each instance then consists out of a p-dimensional feature vector $\boldsymbol{x}_i$ and a corresponding class variable y_i such that $t_i = \{x_i, y_i\}$. At each node, an attribute of the feature vectors is chosen that most effectively splits the instances into their respective class based on the entropy. The entropy is a measure to calculate the homogeneity or purity for each of the resulting subsets. If the subsets can be perfectly divided then the entropy is 0 and when the subsets are completely homogeneous the entropy is 1. The entropy of a (sub)set is calculated with the formula

$$E = \sum_{i=1}^{n} -p_i \log_2 p_i$$

where n is the number of classes in the set and p_i is the relative frequency of class i. Each resulting subset after a split has a different entropy value and the average of these values is called the information gain. To create the average, the entropy is often weighted by the size of each subset. The goal is to find the feature that splits into subsets that maximizes the information gain. The information gain is calculated by subtracting the weighted sum of the entropy of the created subsets from the entropy of the parent set. If that optimal feature has been found, the algorithm is recursively applied on the newly created subsets.

C4.5 (or J48 in WEKA) is an extension of the ID3 algorithm and the predecessor of the newer C5.0 algorithm which is more efficient. Since the C5.0 is patented, the C4.5 is usually implemented. Random Forest differs from C4.5 by using not one but multiple trees. Random forest uses a combination of trees and whereas each tree can have its own training set to increase the classification accuracy. Random Forest initially creates random subsets of instances, whereas the subsets are allowed to overlap. For each subset, a decision tree is generated. A new instance is classified using all decision trees, and the new instance is labeled with the class that has the most recommendations from the random forest. Random forests can be enhanced by bagging. By using bagging, noisy and unbiased models are averaged to create models with low variance. This is done by considering all the features for each node for a split. Decision trees are quite effective in general, and RF compensates for local errors or deviations in the total dataset.

2.4. Performance measures

The method to identify the success of the classification is by using performance measures. Since the hypothesis of this thesis can be formulated as a binary classification problem, there are a few appropriate measures that will be treated here.[5] Estimating the performance of an algorithm is not simply done by looking at the accuracy. The interpretation of the measures greatly relies on the objective. Performance measures can give directions on

[5] https://www.cs.cornell.edu/courses/cs578/2003fa/performance_measures.pdf

tuning the algorithm to the desired prediction. The interpretation of the performance measures is important and will ultimately help in formulating an answer to the hypothesis.

2.4.1. *Confusion matrix*

One of the most important classification concepts is contained in the *Confusion matrix* (or error matrix). This matrix is a table that represents the performance of an algorithm and from which other metrics can be derived. The columns of the matrix represent the instances in a class as classified by the algorithm and each row represents to what class instances actually belong. The matrix makes it easy to see if the system is confusing two classes, hence the name. This matrix can categorize into two categories (e.g. positive and negative) and counts the correctly classified (true) or falsely classified (false) instances per class. Now, a success is when an instance is predicted correctly as a true positive (tp) or a true negative (tn). An error is when an instance's class is predicted incorrectly such that it is either a false positive (fp) or a false negative (fn).

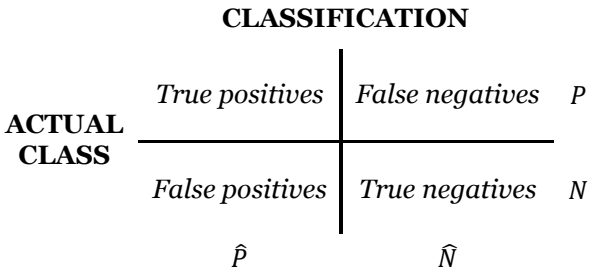

Figure 5 - Confusion Matrix

Because of the topic of this thesis, the formulation becomes a bit counter intuitive. Since deception is often regarded as negative, it would seem logical to consider deceptive instances as negative instances. However, in this thesis the focus is on detecting deception. This means that the deceptive instances are labelled as positive instances and instances coming from truthful messages are labelled negative instances (meaning that they do not originate from deception). This means that a tp classification means a correct classification of the instance originating from a deceptive message. fp are instances that are supposedly truthful but classified as deceptive. fp are instances that were derived from deceptive messages but classified as truthful. tn are instances that originate from truthful messages and are classified as such. The sum of the actual deceptive (positive) instances is P with $P = tp + fn$ and the sum of actual truthful (negative) instances is N with $N = fp + tn$. Then $\hat{P}$ and $\hat{N}$ with $\hat{P} = tp + fp$ and $\hat{N} = fn + tn$ represent the sum of the instances as classified by the algorithm to the respective classes. Now from this matrix, several metrics can be formulated.

2.4.2. *Accuracy*

The accuracy is the total number of correctly classified instances proportional to the total number of instances.

$$Accuracy = \frac{tp + tn}{tp + fp + tn + fn}$$

This is the most popular measure because it yields a value between 0 and 1 where 1 is a perfect classification and 0 is a classification where all the instances were classified incorrectly. A high accuracy does not consequently mean that the classification objective is done. When there are many more negative instances than positive instances, and all the instances are classified as negative, the accuracy alone gives a wrong impression. The interpretation of the accuracy is dependent on the goal of the objective. The inverse of the accuracy is also called the classification error

$$Classification\ error = 1 - Accuracy = \frac{fp + fn}{tp + tn + fp + fn}$$

2.4.3. *Recall and specificity*

Precision and recall are two measures which are often used together to assess the effectiveness of a classification. However, for the classification of truthful and deceptive messages, recall and specificity yield more insight. Consider the instances classified as true positives and false negatives (i.e. all instances originating from deceptive messages). *Recall*, also true positive rate (TPR), is the fraction of correctly classified deceptive

instances divided by all the actual deceptive instances. It can also be described as 'the completeness (quantity) of the results'. The specificity, also named the true negative rate (TNR), is the fraction of correctly classified truthful instances divided by all the actual truthful instances.

$$Recall = \frac{tp}{tp + fn} = \frac{tp}{P}, \qquad Specificity = \frac{tn}{N}$$

When the recall (or specificity) is high, most of all the deceptive (or truthful) instances were correctly classified. When one of the parameters is high and the other parameter is low, the classification is biased as instances of both types are all classified as either deceptive or truthful. This indicates that the classification algorithm is not able distinguish the instances effectively. When both values are low, the classification algorithm confuses the test data and the training data and this indicates that the classification is bad.

Now consider a deceptive message in the test set consisting out keystrokes. If the recall of the classification of the keystrokes is higher than 0.5, then this indicates that more keystrokes from the message were correctly classified than there were incorrectly classified keystrokes. This makes it more probable that the message as a whole is indeed of deceptive intent instead of truthful. If the recall is lower than 0.5, then more than half of the keystrokes (originating from the deceptive message) were classified as originating from a truthful message. The message is then incorrectly classified as truthful. The same reasoning can be applied to truthful messages and specificity. For that reason, these measures are important in the classification of messages for each individual.

3. Methodology

In the previous chapter, a literature review was done to explore the current state of deception detection and keystroke dynamics. A research question will be formulated in this chapter using the insights from the literature review in sub-chapter 3.1. After the research question is formulated, an experiment will be designed in sub-chapter 3.2 to collect the data that can be used to answer the research question.

3.1. Research question

The literature on deception detection shows that people are not good in detecting deception. The accuracy of correctly detecting deception is often measured to be equal to the probability of guessing (Bond & DePaulo, 2006). The accuracy is sometimes even lower due to biasedness. In CMC-environments, deception appears to be equally hard to detect. There are tools available to assert if a person is deceptive, from an obtrusive tool called the polygraph to a very effective voice analyzer. However, there is no tool readily available for a CMC-environment like the Internet, where no visible or verbal clues are present. For plain text messages, people have to assess the messages based on its content.

Based on the research, it is known that deception is a cognitive challenging task. For example, the greater cognitive challenges involved in lying result in longer response latencies and more speech hesitations (DePaulo et al., 2003). It is also known that cognitive challenging tasks have an influence on the typing behavior (Lim, Ayesh, & Stacey, 2014; Vizer et al., 2009). Therefore, the goal of this thesis is to assess if deceiving (making statements up or writing statements with which you deeply disagree in order to comply) as a cognitive challenging task influences the typing pattern.

There has not been a lot of research done on using keystroke dynamics to detect deception. Exactly one recent paper on using keystroke dynamics to detect deceptions has been found (Banerjee et al., 2015). In the study, users were asked to write two essays about a subject. Both and truthful and a deceptive essay had to be written. Using stylometry (i.e. study of linguistic style), the researchers were able to generate a high classification accuracy as high as 80%. The essays also considered other metrics, i.e. the number of backspaces, mouse-up events, arrow keys, the timespan of writing the entire document and the average timespan of writing a word, keystrokes, spaces, non-whitespace keystrokes, and interval between words. Adding these features to the stylometry approach resulted in an increase of the accuracy of a few percent. The researchers stated that for some essays the results were statistically significant and there is some empirical evidence that a change in typing can indicate deceptive behavior. However, looking at the content of a message is privacy invasive. It is therefore useful to look at the effectiveness of keystroke dynamics for deception detection without using the contents of the message.

In order to attain the objectives described in the previous paragraph, the following research question is defined:

RQ: Can keystroke dynamics be used to detect deceptive messages without looking at the contents of the message?

To answer this research question, sub-questions are drafted to structure the research.

The first sub-question describes the incentive to do a literature study on deception detection. Answering this sub-question leads to a justification on why research on deception detection is important and which challenges the field faces. It also yields insights in how an experiment can be designed to lead people to deceive.

S1. What insights does the literature about deception detection yield?

The second and third sub-question describe the incentive to do a literature study on keystroke dynamics. These sub-questions are important to find the characteristics in the data that can be considered as biometrical and that yield different values for a deceptive message and a truthful message. In order to test the hypothesis, it is also important to understand what methods are used to analyze the keystroke data and how a message can be classified as deceptive or truthful.

S2. What characteristics can be extracted from the typing rhythm data that change when people are deceptive and when they are not?

S3. What are the methods that one can use to discriminate between truthful and deceptive messages using only the keystroke data?

Lastly, the corpus from the study of Banerjee et al. is freely available. In order to assess the quality of the acquired dataset, the results can be compared to the results of Banerjee et al. The corpus can be modeled similarly to the PwC dataset and the same classification can be applied to assess the results.

S4. How does the dataset in this thesis compare to the dataset from the other researchers?

3.2. Research design

3.2.1. Conditions

In order to make the analysis as meaningful as possible, it must be carefully considered to what conditions the experiment must comply in order to generate usable data. Preferably, the experiment should be close to a real life scenario. In this sub-chapter, the research design of the data collection phase will be explained.

Various researchers suggested some characteristics that are important to understand media differences. Some of these characteristics are especially germane for deception research, these are: synchronicity, symbol variety, cue multiplicity, tailorability, and rehearsability (Carlson et al., 2004). Synchronicity (or interactivity) pertains to the speed of interaction. Tailorability is the possibility of the medium to adjust the message to individual recipients. Synchronicity and tailorability will add a lot of complexity to the research in terms of design, creation and execution. It is therefore decided that there will be a one-way communication between the assignment (experiment) and the participant. Symbol variety stands for the different symbols with which the communication can take place. Cue multiplicity stands for the number of simultaneous information channels that the medium supports. Since the keystroke dynamics will be used, the symbol variety and multiplicity is limited to the keyboard. And lastly, rehearsability, which refers to the degree of which the medium gives the participant time before and during the interaction to generate a message. It was found that rehearsal reduces arousal induced when lying (Carlson et al., 2004), as the saying 'practice makes perfect' implies. By rehearsal, the individual can think about different possible answers and assess the reliability of each statement. Then he can pick the one that he finds most plausible. By doing so, an individual is better prepared to lie and may have more confidence in his statement than if he would have to improvise. It is therefore important to make sure that the participant does not need to think about what type of answers he is supposed to give on forehand.

The participant should be given some time to adjust to doing an experiment before starting the real experiment. This means that the participant has time to adjust to the keyboard such that changes in the patterns are not just due to the fact that the participant is becoming comfortable. This should not be a large amount of text, otherwise the participant may become fatigued. It should be large enough to become comfortable with the environment.

The typing pattern of each individual is unique to a certain degree. Therefore, the data that is collected per individual must contain samples of deceptive and non-deceptive messages. The participant should not copy a certain text because that will not correctly simulate *normal* typing behavior, but rather his copying ability. Also, the assignments and context in which the samples are typed should be comparable, such that any changes are not due to the fact that the participant changes context.

Literature states that in order to study deception well, the participant should not be indifferent about his lies. White lies can simply go unnoticed whereas other types of deception induce more cognitive effect. The greater the incentive to succeed, the harder he will try to cover up his lies. For example, consider a person sending an email to his employer with a false report of sickness to get a day off. In order to write this mail, the person needs to make up some illness and make it sound credible. It is expected that if the participant has to think more on what he is going to write, this will result in more cognitive strain than when he would just truthfully write that he is in good health and looks forward to going to work. It is expected that this strain will increase when the topic the participant is writing about is more important to the person and his credibility is at stake. This creates a strong incentive to formulate his argumentation as good as possible. In the same manner, it is expected that a participant will be more engaged taking a stance on abortion than when he is defending his favorite color. As the topic becomes more controversial implies that the participant will be more engaged. This can be formulated in a condition. The topic on which the participant has to lie about has to be relevant and important to the participant.

It is known that the typing behavior of an individual changes over time (Lau et al., 2004) and depends on the type of activity or application (Dowland, Fvrnell, & Papadaki, 2002). The experiment should calculate for these variables. This means that the data collection of both deceptive as non-deceptive messages per individual should happen in the same session. Otherwise, a change in typing rhythm may be due to the fact that the typing rhythm changed over time.

Now these conditions can be used to shape the experiment.

3.2.2. Study design

The main goal in this paragraph is to create an experiment where keystrokes can be logged and where the participant types both deceptive and truthful messages. The experiment in this thesis is comparable to the only known existing experiment on keystroke dynamics and deception (Banerjee et al., 2015). In this study, the researchers used amazon Turk to ask participants to write two essays about a subject, a truthful and a deceptive one. The subject was randomly selected to be either gun control, gay marriage or a restaurant review. The main idea in this thesis is to perform a similar experiment and do a more in-depth analysis with more than one truthful and deceptive message per participant and without using stylometry. The researchers concluded that there is indeed a change in pattern, and this thesis will examine how effectively this change can be used to discriminate between truthful and deceptive messages. In this paragraph, the design of the experiment is explained. For the experiment, a website was created that made the participant write four essays, two truthful essays and two deceptive essays. A more detailed description of the design of the website, implementation of the key logger and an assessment of the accuracy of the webserver can be found in Appendix A.

In order to let the participants get comfortable with the keyboard, a few small sample texts are retyped by the participant. By doing so he may forget the activities or things that were on his mind before the experiment. This is also done to let the participant focus on the experiment by giving the participant the feeling the experiment has started. After a sample text is retyped, the experiment will begin. The texts are arbitrary and consist out of 3 sentences.

The participants are then informed about the incentive of the experiment. The study will be transparent in the goal it pursues. This is mandatory because this experiment will take place in the office of PricewaterhouseCoopers. Without background on the experiment some of the questions could be quite sensitive for some actors within PwC. In order to avoid conflicts it was chosen to be transparent about the goal of the research. The participants had to continue by clicking a button with by which they indicated that they agreed with the fact that the recorded data will be used for the experiment. Banerjee et al. also instruct the users on the incentive of the study. Telling the participant on forehand about the incentive of the experiment might be considered a weakness, as the participant can prepare himself or will tend to focus on his typing behavior instead of doing the assignment. But since the participant will be instructed to be deceptive he is expected to understand the goal of the thesis nonetheless.

The assignments on which the participant has to answer deceptive questions can be randomized. In that way the participant cannot prepare himself for answering the questions, which is said to influence the behavior by rehearsability (Carlson et al., 2004). The assignments on which the participants have to deceive should be randomized such that the participants cannot prepare beforehand.

After agreeing with the conditions, the opinion of the participants on four statements were asked. The subjects could agree or disagree with the statement. The statements were all work related and were inspired by the Global People Survey. That is a survey that PwC distributes over his employees to get a general idea of what employees think of their job at PwC. The Global People Survey was just finished by the time this experiment was set up. Based on the survey, the employees have a strong opinion on these subjects given the recent results of the Global People Survey. All statements are connected to their employment and development in the company and question their day-to-day activity. The statements were

1. The brand PwC stands for a high quality of service.
2. PwC knows how to leverage available knowledge and expertise effectively when performing projects.
3. PwC collaborates very well with other PwC offices abroad when performing projects for international clients.
4. PwC is well capable of implementing innovative ideas.

After the true opinions were answered, the participant was asked to motivate his true opinion on two statements. For two other statements, the participant was asked to motivate the opposite of his true opinion on the statement. The statements for which he had to write a statement opposing his true opinion (or writing the deceptive statement) were randomly chosen. For example, consider a participant who answered true for all four statements. That participant would be asked to motivate why he thinks two randomly selected statements are true and he would be asked to motivate why he thinks the other two statements are false. The statements that corresponded with the true opinion and with the opposite opinion were randomly selected. Hence, the participant could not anticipate on which question he was supposed to deceive and on which question he could give his true opinion. The statements had to be at least 100 words long. The choice of words is based on the fact that approximately 1000 characters per truthful/deceptive dataset is needed (Domingos, 2012). There are about 5 letters in a word, so $2\ statements \cdot 100\ words \cdot 5\ characters\ per\ word = 1000\ characters$. This is in line with the study of Banerjee et al, who also required at least 100 words per statement.

After that, the participants were asked to fill in some metadata. The metadata that was collected is browser version, age, sex, self-assessed typing skill and how long they work for PwC. The experiment is further anonymized because the answers should not be traceable back to the participant.

A deviation from the study of Banerjee et al. is the choice to change the subject per assignment. The original study had people write two messages on one topic consecutively. There were two 'flows': writing the truthful text before the deceptive one, and vice versa. A weakness in that approach is that participants can simply negate the statements. For example, consider a participant writing a review for a restaurant he has visited. The participant then poses some arguments, like the food was great. If he then has to write a deceptive review, a review for a restaurant he has not visited, he can actually get influenced by the previous answers. He can simply copy his previous statement or negate it. That way he does not have to come up with new answers because he has mainly already thought about the subject. In this experiment, every statement is work related but every statement is also different from the other. Therefore, the participant has to consider every statement independently. Another difference is that four messages are acquired instead of two.

For a more detailed discussion on the experimental setup, specifically on the weaknesses of the design and the transparency, see the discussion in paragraph 8.2.

3.2.3. Dataset

The data collection took place in the period July 25, 2016 – September 25, 2016. The data was collected on a business unit of PricewaterhouseCoopers. The survey was sent to about 294 people from which 41 people responded. From those respondents, seven were rejected, as the respondents did not finish the assignment properly. Another four responses were rejected, as the participants did not comply by not following up the assignment correctly. Thus resulting in a dataset of 30 participants. Metadata was collected from the participants. The age ranged from 23 to 37 years with an average age of 29 years. From the participants, 24 people were male and 6 people were female. Most participants worked at PwC between 1 and 2 years, all the participants worked at PwC at the time of participation. It was asked how they considered their typing skill and the participants evaluated their typing skills to be as showed in Table 1.

Typing skill	**Bad**	**Below average**	**Average**	**Above average**	**Good**
# of participants	0	0	11	7	12

Table 1 - Distribution of typing skills

The whole dataset contained a total of 66.097 key events which results in an average of 2230 key events per participant and a mean of 550 key presses per message. The distribution of key events for all the messages (4 x 30 participants) are shown in Figure 6, where the x-axis shows the type of message (either truthful or deceptive) and the y-axis shows the number of key events that were recorded while creating the message.

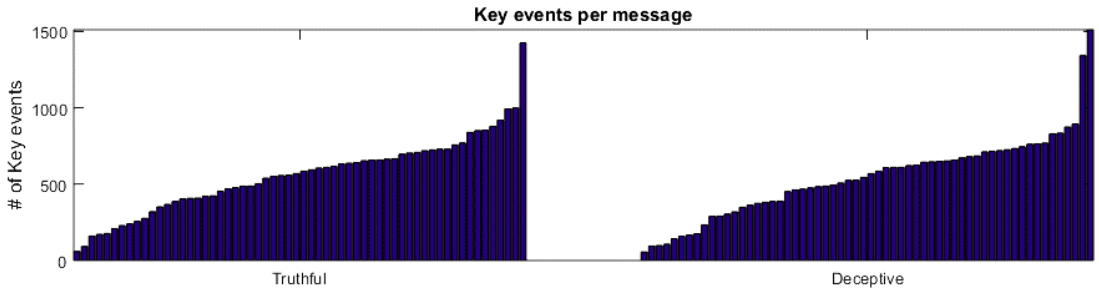

Figure 6 – Distribution of key events per message

The messages were written in an average time span of 153 seconds for a truthful message and 142 seconds for a deceptive message, whereas two extreme gaps needed to be removed for some truthful messages. In Figure 7, again all the messages are plotted with the x-axis shows the type of a message and the y-axis shows the time in seconds between the first and last key event of a message.

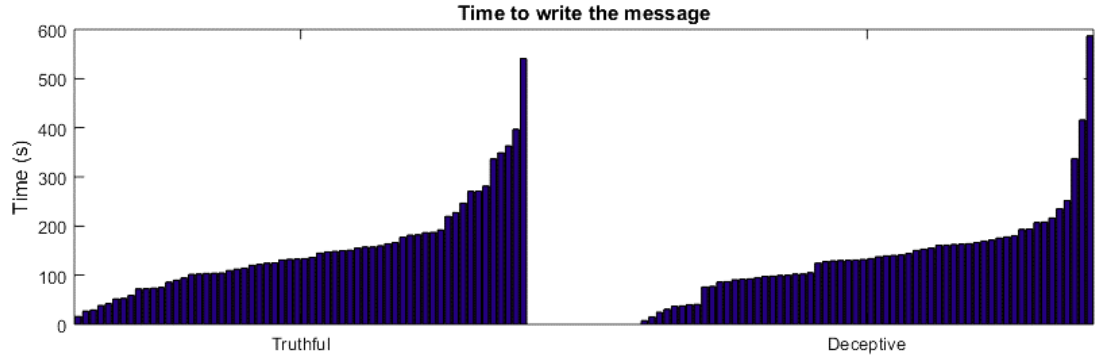

Figure 7 - Writing time per message

In the next chapter, the features that can be extracted from the keystrokes will be treated.

4. Data processing

Data processing is defined as collecting and manipulating the data to produce meaningful information. In order to do so, the data must be validated, sorted, summarized, aggregated, analyzed and finally classified. In this chapter the means by which data is collected, processed and analyzed are presented. There are multiple tools available to analyze the data. The analysis can be done by WEKA, for which the data needs to be transformed into the `.arff` format. The data is first transformed with code written in MATLAB to transform the keystrokes into meaningful features.

In sub-chapter 4.1, some terminology and background on the logged keystroke information is treated. In sub-chapter 4.2, the features that can be derived from the logged keystroke data are explained.

4.1. Keystrokes

4.1.1. Logged keystroke

Data modeling means making the data ready for analysis. In this paragraph, the logged keystrokes are treated. The keystrokes consist of a few variables. These variables are a counter, the char code, the timestamp and the type. The counter is an incrementing integer that decides the order in which the keystrokes are logged. The char code is an integer corresponding to a specific key. A table of the char codes and the corresponding key function can be found in appendix A.4. The timestamp is an integer that corresponds with the exact millisecond a keystroke is logged by the browser. The timestamp can correspond with the internal clock of the computer or with the session time of the browser. The origin is not important as the timestamps are only relevant with respect to other timestamps. The type is a character that defines whether a keystroke describes the pressing (*D*) or the release (*U*) of a key. In Figure 8, a sample of keystrokes can be found with the counter, char code, timestamp and type consecutively.

```
1,16,94392,D
2,79,94482,D
3,79,94576,U
4,16,94596,U
5,78,94638,D
6,78,94740,U
7,84,94770,D
8,84,94850,U
```

Figure 8 - Example of an array of keystrokes

Since the pressing and release of a key always yields two keystrokes, the total number of keystrokes of a message is always an even number.

4.1.2. Key event

A key event is defined as the interaction with a key by pressing and releasing. A key event can be represented by two keystroke lines as described in chapter 4.1.1. These lines have some requirements and characteristics. For a key event, the keystroke with type *D* always comes before the keystroke with type *U*. The character codes of both key strokes of a key event are identical, the types are different and the timestamps also differ. During a key event, other key events may be initiated, as more keys may be pressed and released at the same time. However, for a key event i, the keystroke representing the press comes before the keystroke representing the press of key event $i + 1$. Therefore, the indices of the key events are ordered by the timestamps of the keystrokes corresponding with the key press.

4.2. Features

4.2.1. Choice of features

The field of Keystroke Dynamics describes the behavior of a person. According to the literature, the behavior is described by features which are characteristics extracted from data. In this paragraph, the relevant and useful features are selected from the literature to be used to analyze the behavior and to answer the research question.

According to a survey, the dwell time and the (di-graph) flight time were the most common features with an occurrence of 41% and 49% respectively in 163 reviewed papers on keystroke dynamics (Teh et al., 2013). Another 5% of the papers used keyboard pressure. The remaining 5% of the papers focused on typing speed, typing sequence difficulty, frequency in typing error and sound of typing. The features that will be considered

are the dwell time, (di-graph) flight time, typing speed and frequency in typing error. The typing sequence difficulty will not be considered as it is unpopular and more suitable for password authentication (as often special characters are used). Keyboard pressure and sound of typing will not be considered due to the nature of the experiment. At the end of this chapter, some features such as the pause rate and other quantitative features will be treated that were inspired by the literature on deception.

4.2.2. Dwell time

The dwell time (sometimes called the duration time or hold time) is the amount of time a key is pressed down. It is shown that the dwell time is a consistent metric and changes from person to person (Lau et al., 2004). Let t_i^x be the timestamp of when a key is pressed, let i denote the i^{th} key event and let x be the type with $x = \{down, up\}$ that describes the pressing and release of a key respectively. Let the dwell time d_i be a difference between the timestamps t_i of the pressing and release of the key event i. The dwell time can be written as

$$d_i = t_i^{up} - t_i^{down}, \qquad d_i > 0$$

Many studies use a summary of the dwell time for each possible character, such as creating bins or using the mean and standard deviation (Teh et al., 2013). In order to maintain as much information as possible, the dwell time will not be summarized. The dwell time will be calculated for each key event instead which can then be used for analysis.

4.2.3. Flight time

The second concept focusses on interactions between multiple keys, which consequently yields a lot more features. Consider the interaction between two keys events. The flight time (sometimes called latency) is defined as the time between the press and/or release of different key events. This results in four interpretations of the flight time. It can be the difference between two key presses, two key releases, press-release or release-press of two key events. If two consecutive key events are considered, it is called a digraph. For the difference between two key events with one key event in between, a trigraph. The possible combinations of flight time features grows quickly when key events are allowed to lie between the two key events that are considered. When no key events lie in between (when the permutation possibilities over the press (D) and release (U) of key event i and $i + 1$ are considered) there are 4 possible features combinations. When one key event is allowed to lie in between (such that the permutation possibilities between key event i and $i + 2$ are also considered), another 4 possible features are possible and the total grows to 8 possible features. According to a survey, the di-graph flight time is considered the most effective and is the most popular with a usage of 80% in the studies where flight time is considered (Teh et al., 2013). For that reason, the di-graph flight time is also chosen in this thesis where no key events should lie in between two sequential key events that are considered. Let the flight time be the distance l_i between two consecutive key events i, then

$$l_i = t_{i+1}^x - t_i^x, \qquad x \in \{up, down\}$$

The resulting combinations can be found in Table 2. The Up – Down and Up – Up flight time can become negative, indicating that the latter (second) of two sequential key events is either pressed or released before the release of the former (first) key event respectively.

Name	Notation	Description
Down – Down Digraph	$t_{i+1}^{down} - t_i^{down}$	Strictly positive as $t_{i+1}^{down} > t_i^{down}$
Down – Up Digraph	$t_{i+1}^{up} - t_i^{down}$	Strictly positive as $t_{i+1}^{up} > t_i^{down}$
Up – Down Digraph	$t_{i+1}^{down} - t_i^{up}$	Either positive or negative
Up – Up Digraph	$t_{i+1}^{up} - t_i^{up}$	Either positive or negative

Table 2 - Different combinations to the flight time

Figure 9 shows the dwell and flight time representations for two consecutive key events. These features and their descriptions are based on work done by (Bours, 2012).

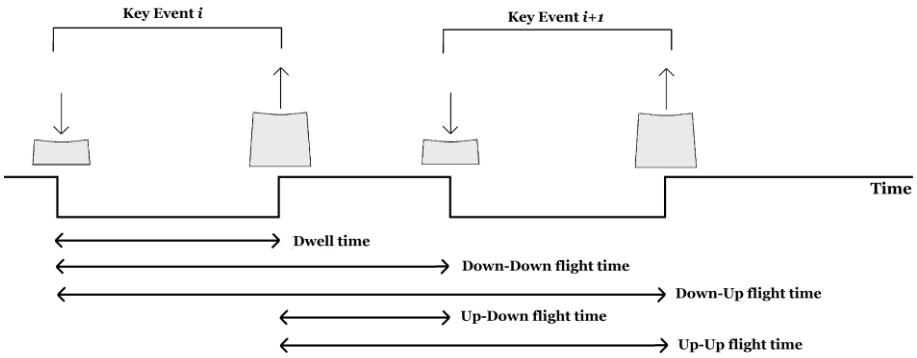

Figure 9 – Dwell time and flight time combinations between two consecutive key events

4.2.4. Typing speed rate

The typing speed rate is the frequency with which keys are pressed. This feature can indicate the flow of a message. The feature can indicate if a person is typing faster or slower. Typing speed changes throughout the message. When the typing speed rate is 0 at a given time, this may also be considered a pause. Typing speed is typically summarized to be used in login security (Revett, Magalhães, & Santos, 2005). Other approaches chose to calculate the typing speed per interval, for example of 1 minute (Kolakowska, 2010). In this thesis, the typing speed is calculated per key event. The typing speed feature can be defined as the number of key events for which the timestamp falls in the interval centered around the current key event. Let t_i be the timestamp in milliseconds of key event i and consider an interval of 1000 milliseconds, then the typing speed ts_i of key event i can be defined as

$$ts_i = \left|\{t_j : t_i - 500 \leq t_j \leq t_i + 500\}\right|, \qquad \forall j$$

Where $|\cdot|$ is the cardinality of the set (i.e. number of elements in the set) and j denotes all the key events coming before and after key event i.

4.2.5. Deletion rate

The deletion rate is a feature describing the number of characters that are deleted. The deletion rate is known to be modeled as a quantitative feature describing whole message, for example a passphrase or a text fragment (Kolakowska, 2010). The effectiveness of modeling this feature per key event (similar to the typing speed), greatly depends on the usage and might therefore be not so useful at it may result in a lot of meaningless elements. In order to understand usage of the deletion keys, a quantitative approach can yield more insight. Therefore, a quantitative approach is chosen for the exploratory analysis.

4.2.6. Pause rate

The pause rate is a measure for the time in which there is no interaction with the keyboard while writing a message. The pause rate does not appear as a common feature in the literature. However, since deceiving is a cognitive intensive process and the literature has shown that deceivers often display longer pauses in their responses (Vrij, 2008), this feature could yield useful insights.

The pause rate is a measure for the time between two key events. Technically, this feature resembles a special case of the flight time (i.e. Up – Down or $\{t_{i+1}^{down} - t_i^{up} : t_i^{up} < t_{i+1}^{down}\}$). As has been mentioned earlier, key events may overlap resulting in a negative flight time. A key event also focusses on typing whereas the absence of key events is considered a pause. Given that pauses are not ubiquitous during typing of a word, it is more relevant to look at the pauses between the writing of words than the pauses between key events. Each time a word is written, the participant may take a moment to think, where the pauses may be different depending on the intent of the message.

4.2.7. Other quantitative features

Despite the fact that there are no other obvious features apparent in the literature, it was shown that deceivers are more hesitant in conversations and display smaller response lengths. These quantitative features may be considered by looking at the length of the message, the deletion rate (see 4.2.5) and the writing time.

5. Exploratory data analysis

In this chapter, the data is analyzed in order to find promising approaches that will help in the classification of the data. Clustering is the main task of exploratory data mining. Clustering is grouping a set of objects in clusters such that objects in the same clusters are more similar to each other than those in other clusters. In order to find these similarities (in this thesis called features), the data is studied in detail. This chapter enhances the intuition about the data and helps in finding the best clustering parameters for the data. The research that has been done on DD and KD uses high-level features and there is a scarcity in detailed analysis of the data. In order to analyze the data, the features that were introduced in the literature are studied. The descriptions of these features can be found in chapter 4.2. The PwC dataset is used which consists of 120 messages written by 30 employees (i.e. 4 messages per employee). The features will be analyzed for each participant individually, as keystroke dynamics is a biometric in sub- chapter 5.1 and 5.2. The PwC dataset will be compared to the dataset of Banerjee et al in sub- chapter 5.3. Finally, in sub-chapter 5.4 the features that will be used in this thesis will be chosen.

5.1. Analysis of key events

5.1.1. Approach

In this paragraph, the basic features that are often presented in literature are treated. These features are the dwell time, di-graph flight time and the typing speed. All these features can be calculated for all key events except for the last key event as there is no consecutive key event. Since each participant may exhibit different behavior and given the fact that (typing) behavior is unique from person to person, the analysis is done for each participant individually. That means that the truthful messages need to be compared to the deceptive messages of each participant. If the typing behavior is different according to the feature, then this might indicate that discrimination of the two types of messages is possible for that person and feature. Consequently, this means that the appropriate set of features may be different for each user.

In order to assess the differences between the two types of data (features coming from either the deceptive messages or the truthful messages), the Probability Density Function (PDF) is used. The PDF displays with what probability a certain value of the feature appears in the message. The values the feature can attain are plotted on the x-axis while the probability is plotted on the y-axis. The PDF can be used to visualize and calculate the differences of each feature between the two types of messages. The absolute difference in area between two PDFs can be calculated for each feature, by plotting a PDF of both the deceptive and truthful message in the training set of each user. This yields a difference value that reflects the difference of the feature between the truthful and deceptive training set. A PDF has an area that is equal to 1. In order to get the best insight on the differences between two PDFs, the absolute value of the difference over the whole domain are added up. The lower bound of the difference is 0, when the PDFs perfectly overlap. The upper bound for the difference between two PDFs is 2, which occurs if the two PDFs do not overlap at all. In Figure 10, a plot of two PDFs is shown. In the plot, the marked area represents the difference between the two PDFs. The advantage of this measure is that the differences for frequent and incidental feature values adjust their significance on the measure according to their occurrences. This means that differences in frequent values have a larger impact on the difference measure than infrequent values.

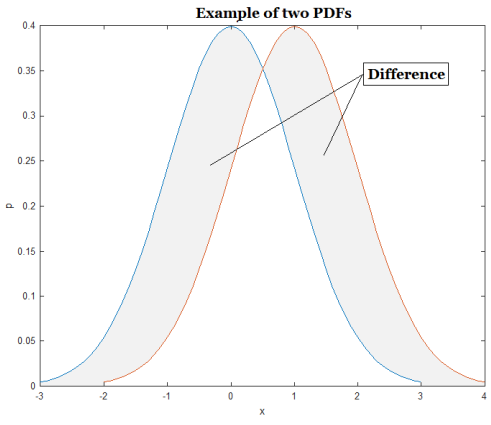

Figure 10 - Example of two PDFs where the grey area represents the difference

5.1.2. *Dwell time*

The dwell times can be plotted on the x-axis and the y-axis represents the probability of the dwell time. The PDFs of the dwell times of the two message types of participant 2 are shown in Figure 11. The plots of the deceptive and truthful data look quite similar. The corresponding difference between the PDFs is roughly equal to 0.17, which can therefore be considered low.

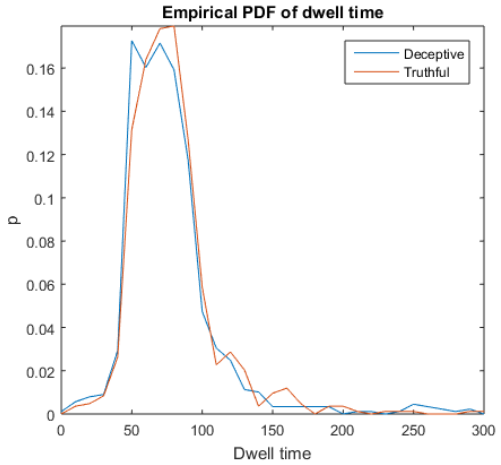

Figure 11 - Empirical PDFs for dwell time per message type of participant 2

Given the plot and the difference measure, it can be decided if the dwell time is an appropriate feature for this person. The difference measures for all users is given in Table 5. The mean difference for all users is 0.25. This is in contrast with the statistical tests, which showed statistical significance for 9 participants. The dwell time was eventually not chosen for any participant, which will further be treated in chapter 5.4.

5.1.3. *Flight time*

There are four different types of di-graph flight times. The PDFs can be plotted to see differences between the two types. The plots for participant 2 are shown in Figure 12. The difference values of the PDFs in this plot are 0.37, 0.36, 0.33 and 0.33 for the Down-Down, Down-Up, Up-Down and Up-Up flight times respectively. All the plots confirm that there are differences in both the PDFs. The range of values the flight-time attains is much larger than for the dwell time. The PDFs generally shown that there are more high-valued flight times for the deceptive messages than there are for the truthful ones. The average differences between the PDFs for all types of flight times lies between 0.44 – 0.45. Although the statistical tests showed that there were fewer participants for which the differences were statistically significant than there were for dwell time, the feature was adopted the most in the feature sets. As will be shown in the features selection further on, the different variants of the flight times were used in approximate 95% of the cases.

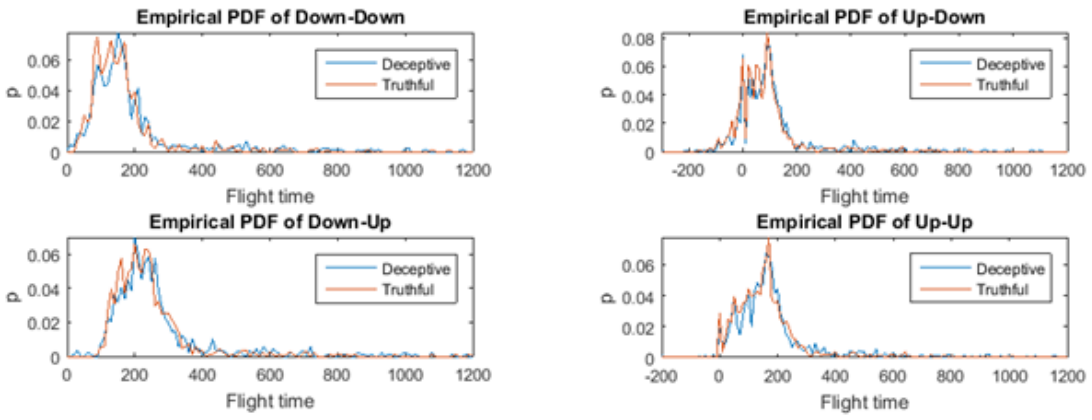

Figure 12 – Empirical PDFs of the four flight times and the two message types for participant 2

5.1.4. *Typing speed*

In this paragraph, the typing speed is treated. The typing speed per key event is the number of key presses that occur within 500 milliseconds of the key event. The fastest typing speed while writing a messages was 30 key presses per second, which was achieved by holding a key longer than 500 milliseconds. This behavior most often occurred with the shift key, as this key is often pressed when starting a sentence. These high typing speeds are not considered in this analysis because they do not reflect the true typing speed the participant performs. Without considering the long key presses, the average typing speed does not exceed 10 characters per second for each user. In Figure 13 the PDFs are plotted of the typing speed of deceptive and truthful messages for participant 2. The difference value for these PDFs is 0.2575.

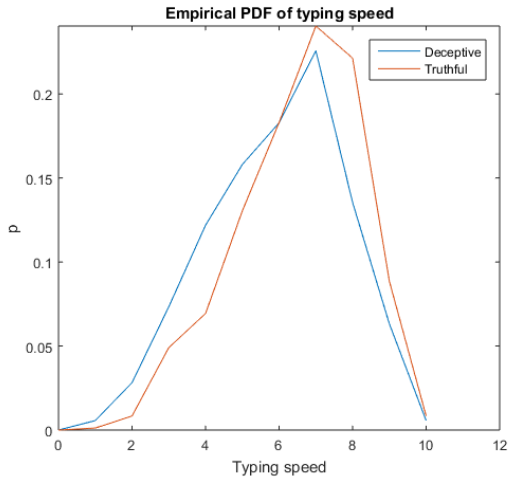

Figure 13 - CDF of typing speed of the two types of messages of participant 2

The typing speed showed the highest number of participants for which the differences were statistically significant. The feature was chosen for about 4 participants.

5.1.5. *Statistical difference*

In order to assess the differences in the data, a statistical test is done. This test will not be used for feature selection, as both the training and test set are used but give insight in the data. Also, the statistical test can be compared to the dataset of Banerjee to see if there are significant changes. The statistical test that will be used is the Mann-Whitney U-test, a test often used to find out if two signals come from a different source. In Table 3, the results of the statistical tests are shown. When a p-value of 0.05 is set as a threshold, the number of participants with a statistical significant difference are 9, 7, 9, 3, 3 and 13 for dwell time, down-down time, down-up time, up-down time, up-up time and typing speed respectively. The features for which the differences are statistically significant are denoted with asterisk (*). Although insightful and possibly hinting on the outcome of the results, the statistical significance will not be used in choosing the features. The difference in significance may be due to the fact that the flight times can attain very high (and low) values, e.g. the domain is wide. This increases the standard deviation which clearly has an influence on the results.

	Power of test (U)	Dwell time (p-value)	Down – Down (p-value)	Down – Up (p-value)	Up – Down (p-value)	Up – Up (p-value)	Typing speed (p-value)
1	728696	0,45	0,79	0,60	0,93	0,87	0,95
2	370348	0,04*	0,00*	0,00*	0,00*	0,00*	0,00*
3	770182	0,02*	0,01*	0,00*	0,25	0,06	0,00*
4	451863	0,61	0,32	0,22	0,57	0,37	0,75
5	712185	0,40	0,14	0,07	0,10	0,07	0,00*
6	1188750	0,17	0,08	0,04*	0,37	0,25	0,14
7	137326	0,67	0,98	0,92	0,89	0,95	0,30
8	377363	0,94	0,76	0,79	0,57	0,70	0,73
9	741744	0,82	0,51	0,80	0,85	0,83	0,39
10	493350	0,13	0,09	0,29	0,12	0,38	0,00*
11	953019	0,06	0,00*	0,00*	0,01*	0,00*	0,00*
12	471744	0,45	0,94	0,96	0,89	0,92	0,28
13	673221	0,89	0,75	0,71	0,92	0,80	0,24
14	833112	0,13	0,02*	0,00*	0,22	0,12	0,00*
15	504444	0,57	0,66	0,65	0,83	0,75	0,89
16	837188	0,18	0,45	0,31	0,52	0,38	0,03*
17	675193	0,09	0,04*	0,02*	0,14	0,08	0,00*
18	17685	0,01*	0,05	0,02*	0,23	0,14	0,01*
19	364161	0,00*	0,91	0,26	0,33	0,88	0,21
20	1277276	0,04*	0,75	0,76	0,47	0,71	0,05
21	238602	0,01*	0,67	0,24	0,47	0,96	0,71
22	793788	0,03*	0,26	0,01*	0,63	0,61	0,00*
23	416185	0,30	0,92	0,70	0,58	0,95	0,32
24	163011	0,05	0,04*	0,14	0,01*	0,04*	0,00*
25	846299	0,72	0,10	0,07*	0,27	0,16	0,08
26	710525	0,00*	0,98	0,18	0,50	0,51	0,67
27	99408	0,57	0,91	0,82	0,68	0,92	0,64
28	1196443	0,07	0,02*	0,00*	0,44	0,13	0,30
29	401520	0,65	0,27	0,10	0,52	0,34	0,01*
30	28091	0,01*	0,60	0,33	0,88	0,56	0,02*

Table 3 - Mann-Whitney U-test for the key event features

5.2. Analysis of specific interactions

5.2.1. Introduction

In this paragraph, some features are studied that cannot be modeled per key event. The literature implies that lying is a cognitive intensive process and it is shown that deceivers display longer pauses and lower response lengths consequently (Vrij, 2008). In this chapter, these characteristics are treated and modeled into features to be reviewed. Figure 14 shows the char codes (on the y-axis) plotted against the timestamps of all key events of a message. Each circle represents a key press. The arrows point to different places in the message where deletion, typing or a pause occurs. These figures are insightful to generate an idea on what happened while the participant was writing the message, but visually analyzing the 120 messages did not yield a new insight.

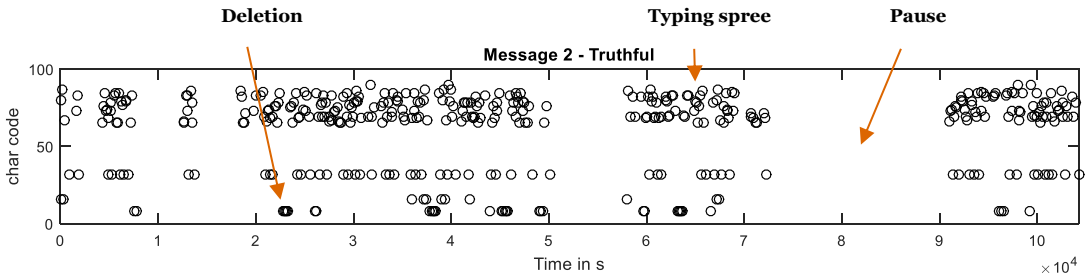

Figure 14 – Time series of categorized key chars of a participant

The deletion keys also show a certain pattern. For small corrections, the key is used one time. In a lot of cases however, the key is used frequently to delete a large amount of text. It would be useful to recognize this behaviour and to find out if the frequency and length with which these deletion moments occur is influenced by the intent of the message. Furthermore, the typing speed or pause rate may indicate the thoughts a user has put in while writing a message. Writing a deceptive message could supposedly cost more time as the participant has to come up with arguments. Or maybe will be done faster as he is proficient with lying. That is what will be analysed in the coming paragraphs.

5.2.2. Quantitative message properties

In this paragraph, some quantitative properties between the deceptive and truthful messages will be analysed. The properties are message length, time to write the total message and the deletion rate. These aspects are derived from the literature on deception, which states that often these traits also change for real-life deceivers (Vrij, 2008).

First, the message length is analysed. Deceivers are said to produce shorter answers. This might be reflected in the number of key events per message. In Figure 15, the number of key events are plotted per message type and participant (x-axis). The figure indicates that there is no quantitative pattern visible that discriminates truthful from deceptive messages. The number of key events per message seem distributed randomly. It is shown that each participant has messages of equal size. The number of key events does not visually indicate cues that discriminate deception from truthfulness.

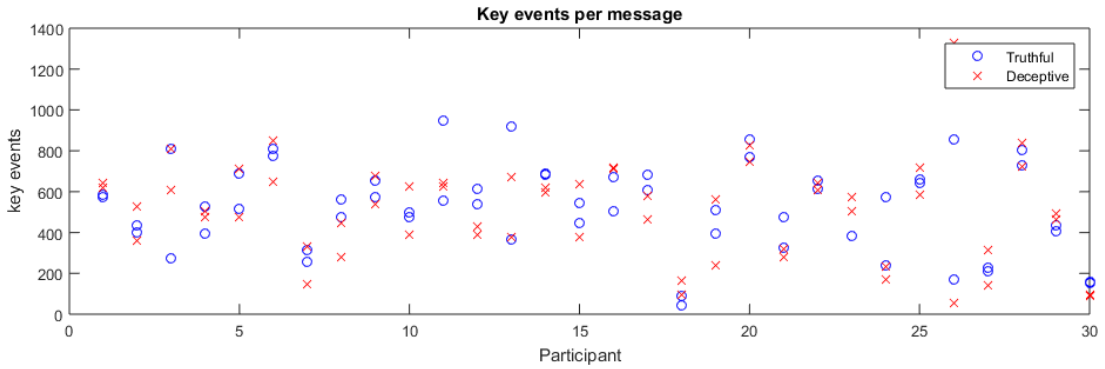

Figure 15 - Key events per message of each participant

Key events do not reflect the length of the message properly. The number of key events may increase as a user presses the shift or delete, but this does not change the size of the semantical message consisting of alphabetic characters. The true number of alphabetic characters of each message can be plotted against the number of key events that was needed to form this message, to see the differences between the messages per intent. In Figure 16, the true message length is plotted against the number of key events. The plot does not show a difference between the two types of messages, as both the truthful and deceptive messages are equally distributed throughout the range.

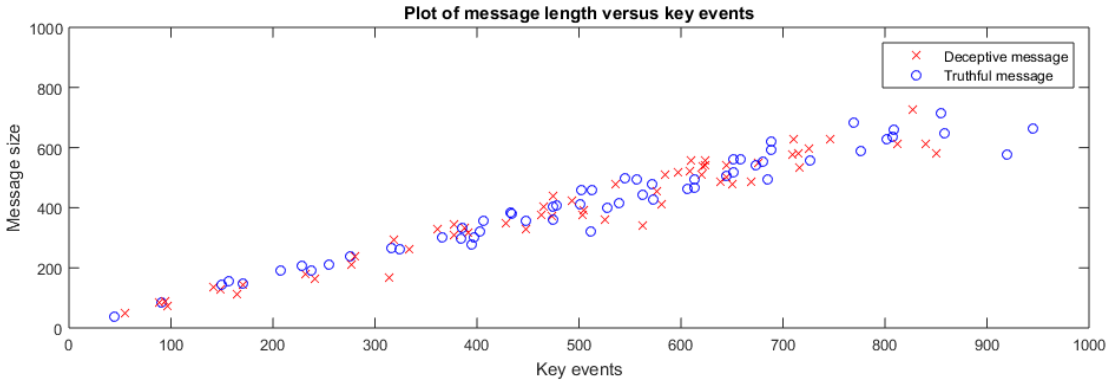

Figure 16 - Plot of true message length against the number of key events

Now the literature on deception also showed that deceivers need more time to formulate an answer (Vrij, 2008). In Figure 17, the writing time in seconds is plotted for each message per user. The figure shows that writing time does not obviously discriminates messages by intent per person. The messages are distributed rather homogeneously which will not favour a classification job based on this metric.

Figure 17 - Plot of the writing time in seconds for each message per user

Lastly, the number of deletions can be plotted against the total number of key events. Deletions may have a predictive force when it comes to deception, as participants may be more hesitant about their writing when they are deceptive. In Figure 18, the number of deletions is plotted against the key events of the corresponding message. It is shown in the plot that the truthful and deceptive messages are spread homogenously. This makes deletion per length of message unfit for classification.

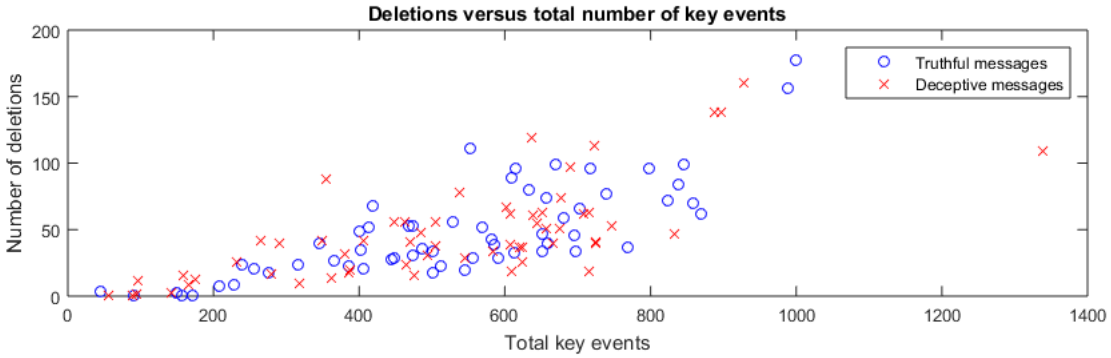

Figure 18 - Number of deletions against the total number of key events of a message'

To conclude, the features considered in this sub-chapter do not show visible differences that are suitable to be used for classification.

5.2.3. Pauses between words

After writing each word, the participant may take a pause. During these pauses, the participant may do several things. He may continue writing, take some time to think, look for spelling errors, or check the contents of his written words. The pauses during these breaks might be indicators to deception. In order to analyse these pauses, each part in the text where two words are separated is analysed. This separation is typically characterized by a space. In Figure 19, the PDFs of the messages of participant 2 are plotted. The plot does not show obvious differences between the two messages, with a difference value of 0.3744.

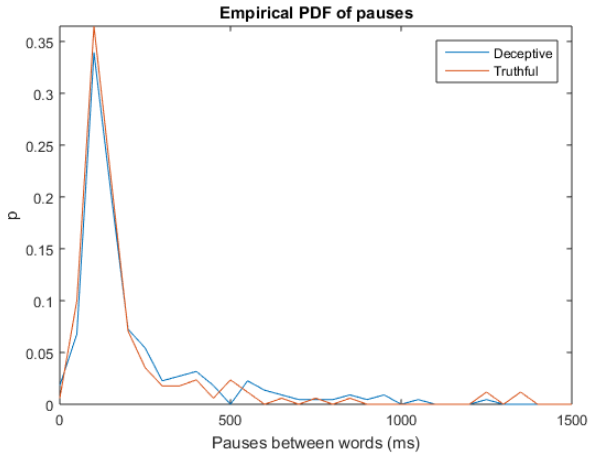

Figure 19 - Pauses between words per message type of participant 2

5.2.4. Statistical differences

The quantitative features, as displayed in chapter 5.2.2, show no visible difference between the two types of messages. The pause rate however, did show some differences. In this sub-chapter, the statistical properties of the pauses between words of the two message types for each participant will be calculated. For the statistical test, a Mann-Whitney U-test will be used. The results are shown in Table 4. A p-value of 0.05 is chosen. The number of participants with statistical significant differences in the pause rate between words is 1.

	Power	**Pause rate (p-value)**
1	16650	0,87
2	7497	0,92
3	18785	0,20
4	8512	0,56
5	17391	0,36
6	23618	0,33
7	3176	0,66
8	5895	0,36
9	13448	0,06
10	11020	0,22
11	19292	0,00*
12	10034	0,85
13	12740	0,66
14	19623	0,76
15	11250	0,93
16	17922	0,33
17	13659	0,27
18	441	0,50
19	6825	0,25
20	29507	0,17
21	5792	0,13
22	18236	0,17
23	7473	0,10
24	3328	0,25
25	16888	0,76
26	14453	0,92
27	1922	0,90
28	22038	0,79
29	8954	0,89
30	528	0,08

Table 4 - Mann-Whitney U-test for the pause rate

5.3. *Analysis of dataset of Banerjee et al.*

In order to assess the quality of the PwC dataset, the features of the PwC dataset will be compared to the corpus (dataset) of Banerjee et al., which is freely available[6]. The dataset was created using Amazon Turk, an online platform where participants can do experiments to be used for AI research in exchange for a small amount of money. The corpus consists out of three datasets. The first dataset was created by participants who wrote both a truthful and false restaurant review. The second and third dataset was created by letting participants write a small essay about their stance on gun control and gay marriage. The data was labelled according to the stance on the subject, which was answered initially before writing the essays. The datasets consist out of 1000, 800 and 800 texts respectively where each unique participant wrote 2 reviews/essays, both truthful and deceptive. Each unique participant contributed to only one dataset.

For a good comparison, the gun control dataset is chosen. The design of the experiment comes close to the design in this thesis and the classification results were good. Because the dataset was created by 400 participants, the data is sampled to 100 participants whom are randomly chosen. The features as treated in this thesis, are calculated for each participant and their (deceptive and truthful) messages.

5.3.1. Statistical difference

In order to compare the dataset to the PwC dataset, a statistical test is done. The statistical test that is used is the Mann-Whitney U-test. The results (U- and p-value) are shown in Appendix B.2. With a threshold of $p < 0.05$, the number of participants with statistical significant features are 34, 29, 31, 23, 19 and 53 for the dwell time, down-down time, down-up time, up-down time, up-up time and typing speed respectively. The threshold for the pause rate is also chosen to be $p < 0.05$. This threshold resulted in a statistical significant difference in the pauses between words between the two types of messages for 14 participants.

5.4. *Feature selection*

In this chapter the features that can be used for classification were studied. In chapter 5.1, the most common features were studied per key event. The features were dwell time, flight time and typing speed. In chapter 5.2, the quantitative properties of the messages were explored such as the message length, time to write the message and deletion rate. It was found that these quantitative features do not possess predictive characteristics that can be used for distinguishing truthful from deceptive messages. The pauses between words were also studied for each participant and between the message types.

By looking at the PDFs and calculating a difference value, the suitability of a feature for each participant can be assessed. The results of the difference in values for each participant can be found in Table 5. In order to select the best features, a best-of-three approach is taken. That means that the three features with the highest difference value compared to the other features are chosen. The pause rate is modeled in a different way than the other features. Since there are fewer words than key events, there are consequently fewer pause rate instances. For that reason, the pause rate will be considered different from the feature set consisting out of the three features that are selected from the dwell time, flight times and typing speed. In paragraph 5.4.1 the PwC dataset will be considered and in paragraph 5.4.2 the dataset of Banerjee et al. will be considered.

5.4.1. PwC dataset

The PDFs of the messages per participant were analyzed and a difference rate was calculated. In Table 5, the difference values for the pauses between words can be found for each participant. The features annotated with an asterisk (*) are the features with the highest difference value and are adopted in the feature set of that participant.

It is shown in the table that adoption rate for the dwell time is non-existent, as no participants had a higher difference value than other features. The adoption of the typing speed was higher, with 4 participants having a large enough difference value*. The flight time appears as the most effective discriminating feature, with an adoption rate of 18, 24, 19 and 25 for the Down-Down, Down-Up, Up-Down and Up-Up respectively.

[6] http://www3.cs.stonybrook.edu/~junkang/keystroke/

Participant	Dwell time	Down-down	Down-up	Up-down	Up-up	Typing speed	Pauses
1	0,29	0,33	0,45*	0,36*	0,36*	0,16	1,05*
2	0,20	0,42	0,52*	0,46*	0,45*	0,28	0,79*
3	0,20	0,25	0,26*	0,29*	0,32*	0,20	0,59*
4	0,28	0,37	0,40*	0,38*	0,40*	0,15	1,12*
5	0,24	0,33	0,38*	0,38*	0,36*	0,20	1,06*
6	0,16	0,37*	0,39*	0,41*	0,36	0,15	0,99*
7	0,47	0,65*	0,66*	0,59	0,67*	0,22	1,42*
8	0,25	0,43	0,55*	0,48*	0,53*	0,28	1,44*
9	0,15	0,39*	0,35	0,41*	0,39*	0,26	1,20*
10	0,13	0,48*	0,45*	0,43	0,46*	0,25	1,11*
11	0,22	0,37	0,42*	0,34	0,41*	0,38*	1,03*
12	0,22	0,40	0,46*	0,40*	0,43*	0,33	0,98*
13	0,25	0,32	0,38*	0,36*	0,34*	0,25	0,95*
14	0,12	0,40*	0,39*	0,37	0,31	0,43*	0,67*
15	0,21	0,46*	0,40*	0,37	0,45*	0,16	1,21*
16	0,19	0,30*	0,40*	0,29	0,37*	0,14	0,91*
17	0,19	0,40*	0,46*	0,39	0,40*	0,39	0,97*
18	0,67	1,22*	1,29*	0,96	1,10*	0,43	2,00*
19	0,27	0,59*	0,58*	0,54	0,57*	0,13	1,12*
20	0,15	0,34*	0,35*	0,35*	0,30	0,19	1,12*
21	0,24	0,49*	0,45*	0,41	0,49*	0,13	1,12*
22	0,28	0,31*	0,30	0,32*	0,39	0,26	0,79*
23	0,17	0,33	0,36*	0,44*	0,41*	0,20	1,38*
24	0,22	0,54*	0,44	0,47*	0,47*	0,40	1,27*
25	0,21	0,34*	0,33	0,42*	0,34*	0,13	1,01*
26	0,42	0,33	0,35*	0,45*	0,41*	0,22	0,99*
27	0,41	0,56	0,56*	0,51	0,57*	0,66*	1,25*
28	0,21	0,42*	0,41	0,42*	0,46*	0,16	1,01*
29	0,18	0,36*	0,33	0,35*	0,34	0,45*	1,12*
30	0,39	0,67*	0,67*	0,71*	0,58	0,19	1,41*

Table 5 - Difference measure of the PDFs for each user and feature

5.4.2. The Banerjee et al. dataset

The feature selection procedure for the Banerjee et al. dataset is similar to the feature selection of the PwC dataset. The PwC dataset has the advantage that there are two deceptive and two truthful messages per participant available, whereas the Banerjee et al. dataset only has one message per type available. In order to separate the training and test set, the messages are divided in half. The first quarter and third quarter of a message is used to select the features and for training, whereas the second quarter and fourth quarter of the message is used as the test set. The features showed some strange behavior for some users. Figure 20 shows the first two key events of the fake essay of participant 1563 on gun control. The dwell time for the first key event is 8 milliseconds, which is very low compared to other participants. However, the flight time seems normal.

```
1375976665344 KeyDown   72
1375976665352 KeyUp     72
1375976665961 KeyDown   79
1375976665971 KeyUp     79
```

Figure 20 - First two key events of participant 2

The difference values for the features can be found in appendix B.3. The dwell time was non-existent using the best-of-three method, and was not chosen for a particular feature set. The typing speed was selected for 7 participants. The flight time had the highest selection rate with an adoption for 63, 82, 64 and 84 participants for the down-down, down-up, up-down and up-down flight time respectively.

6. Data analysis

Now that the data is modeled and the features are extracted from the data, the next step is to produce a meaningful analysis by using methods as selected in chapter 2.3. In sub-chapter 6.1, the approach to analyzing the dataset will be explained. In sub-chapter 6.2, the distance based classification will be considered. In sub-chapter 6.3, classification of the PwC dataset using machine learning methods will be considered. Finally, in sub-chapter 6.4, the classification of the Banerjee et al. corpus using machine learning will be considered.

6.1. Approach

6.1.1. Datasets

In the PwC dataset, each participant yields one complete dataset consisting out of two truthful and two deceptive messages. The PwC dataset is naturally divided into four subsets, i.e. the four messages with the corresponding intent. The dataset of Banerjee et al. contains only one deceptive and truthful message per participant. The analysis will be done using a distance based approach and using machine learning. For each of these approaches, the data is split differently.

First the distance based classification method is considered. Both the training and test sets of both datasets are 50% of the total, resulting in a 50/50 division. For the PwC dataset, two messages (one truthful and one deceptive) are randomly selected to be used for training. These are the same messages for which the features were selected. The remaining two messages will be used as the test set. The training set is used to adjust the classification method parameters and the test set is used to validate the effectiveness of the classification method to new data. This yields the division as shown in Figure 21. Since each message consists of approximately 500 key events, both the training and test sets will consist of an average of 1000 key events per participant. The training set of the Banerjee et al. dataset consists of the first and third quarter of each message. This is the same split for which the features were selected. The second and fourth quarter of the message will be used for validation (test set). For the classification, both the feature set and the pause rate as discussed in 5.4 will be used.

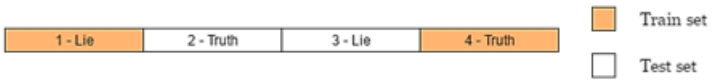

Figure 21 – Exhaustive 2-fold cross-validation

The WEKA experimenter uses other means to split the data. This means that for the machine learning methods, a different kind of split will be used. A cross-validation approach with 4 folds and 5 iterations is used. This approach can be used for both the PwC dataset as well as for the Banerjee et al. dataset (in their paper, a 5-fold cross validation approach was used for classification). The pause rate will not be considered for machine learning as the pause rate is a 1-dimensional feature which is not appropriate for the algorithms. Only the feature set will be tested. Samples from the results

6.2. Distance based classification

6.2.1. Classification of the PwC dataset

The distance based classification is calculated by the sum over the feature vector of the absolute difference to the mean of the test set normalized by the mean average deviation of the test set. In chapter 5.4, a set of features was selected for each user. After calculating the mean vector and mean absolute deviation vector for both types of the training set per participant as explained in 2.3.1, the classification performance can be evaluated for each instance. When the distance to the truthful training set is smaller than the distance to the deceptive training set, an instance is considered truthful and vice versa. If more instances of a message are closer to the truthful dataset, the whole set of instances is considered a truthful message and vice versa.

The results of the classification are presented in Table 6, which represents the total 60 messages in the PwC test set. The top row depicts the actual deceptive messages and the bottom row depicts the actual truthful messages and each row sums to 30 for each type of each participant. The left column depicts the messages that are classified as deceptive and the right column depicts the messages that are classified as truthful. For one participant, both messages were classified wrong. For another participant, both messages were classified correctly. For each of the remaining participants, one message was incorrectly classified and one message was correctly classified.

	Classification	
Actual Class	18	12
	18	12

Table 6 – Confusion matrix for the classification using the key event feature sets

Since the features of the pause rate cannot be modelled per key event, this feature will be treated separately. The distance-based classification method yields the classification results as displayed in Table 7. For two participants, both messages were classified wrong. For three participants, both messages were classified correctly. For each of the remaining participants, one message was incorrectly classified and one message was correctly classified. The classification shows the same classification bias as for the classification using the feature set.

	Classification	
Actual Class	18	12
	18	12

Table 7 - Confusion matrix for the classification using the pause rate

The individual classification results per participant can be found in Table 8 where each row represents one participant. The table shows the tp, fp, tn and fn of the key events coming from the two types of messages in the test set. Choosing the largest number to classify the message (tp aginst fp and tn against fn) from each test set type is chosen, the results as in Table 6 and Table 7 will appear. Ideally, when one of the two classification approaches (feature set or pauses) performs badly, a combined classification rate can be calculated to enhance the performance. In this case however, the feature set and the pause rate show the same bias. Combining the accuracies does not result in a better performance. In the cases where either the feature set or the pause rate performs better than the other, combining the accuracy would result in a decrease of the accuracy instead of a rise as the performance overall is bad.

	Key event feature set				**Pauses**			
Participant	**Deceptive**		**Truthful**		**Deceptive**		**Truthful**	
1	**TP**	**FP**	**FN**	**TN**	**TP**	**FP**	**FN**	**TN**
2	192	452	183	399	15	63	21	67
3	81	444	46	387	8	48	7	60
4	202	610	75	201	38	89	12	31
5	218	286	183	214	32	29	23	32
6	98	377	123	566	21	86	19	82
7	472	378	436	340	66	52	57	62
8	151	182	112	143	15	35	15	35
9	433	15	458	20	26	10	51	9
10	300	236	323	249	85	6	77	7
11	323	301	258	243	35	19	52	26
12	338	286	420	525	69	17	98	34
13	259	169	364	249	57	1	93	0
14	337	332	520	400	52	3	51	1
15	243	378	337	343	40	49	35	70
16	269	370	185	263	20	34	21	46
17	389	327	391	282	62	46	61	36
18	247	218	363	322	32	40	48	43
19	160	5	86	5	7	7	4	2
20	353	209	300	212	29	3	49	10
21	326	420	281	488	6	125	5	128
22	189	129	177	147	49	1	44	5
23	514	96	544	69	65	24	68	25
24	288	217	222	162	35	35	29	19
25	197	35	197	41	26	0	37	1
26	497	218	458	200	71	23	78	12
27	18	37	332	526	51	135	30	90
28	142	172	136	72	19	2	33	3
29	434	406	409	393	99	16	99	15
30	261	232	247	187	61	10	42	19

Table 8 - Confusion matrices of all the key features per user

6.2.2. *Classification of the Banerjee et al. dataset*

The PwC dataset can be compared with the Banerjee et al. dataset. The confusion matrix of the distance based classification using the feature sets is presented in Table 9. There was one participant for which both parts of the messages were classified incorrectly. For another 5 participants both the parts of the messages were classified correctly. For the remaining 94 participants both messages were either classified as deceptive or truthful.

	Classification	
Actual Class	56	44
	52	48

Table 9 - Confusion matrix for the classification of messages using the key event feature sets

The confusion matrix of the distance based classification using the pause rate is presented in Table 10. There were two participants for which both the parts of the messages were classified correctly. For the remaining 98 participants both messages were either classified as deceptive or truthful.

	Classification	
Actual Class	51	49
	49	51

Table 10 - Confusion matrix for the classification of messages using the pause rate

The results and a comparison will be further treated in chapter 7.

6.3. *Classification of PwC dataset using machine learning*

In this sub-chapter, the algorithms which will be used for classification are tuned. Afterwards, the PwC dataset is classified using the algorithms and the classification performance is presented. A test message is classified as truthful or deceptive if the specificity (true negative rate) or sensitivity (true positive rate) of the classification exceeds 0.5 respectively. This means that more key events of a message are classified correctly then incorrectly, making it more probable that the message is of the specific type.

6.3.1. *Naive Bayes*

The are no relevant parameter that can be adjusted for the Gaussian Naive Bayes (the prior class probability is not suitable for this task). In Table 11, the confusion matrix together with the important performance indicators is presented. The classification is done using 4-fold cross-validation. The table shows the average accuracy and standard deviation over all 30 participants from the PwC dataset. For all 30 participants, both messages were either classified as truthful or deceptive. The sensitivity and specificity do not reflect properly what happened for each participant individually, as each of the metrics are summaries of the 30 participants. The standard deviation is quite high, implying that either the sensitivity or specificity was quite high and the other one quite low. And for other participants, this was the other way around. Therefore, summaries do not reflect properly what is actually happening. The confusion matrix together with a description for how many people the classification bias occurred is a better approach.

	Classification	
Actual Class	17	13
	17	13

	μ	σ
Accuracy	0.50	0.06
Sensitivity	0.56	0.44
Specificity	0.44	0.48

Table 11 - Confusion matrix and performance indicators for NB for the feature set

The NB implementation in WEKA is able to deal with continuous data, as the program uses a Gaussian distribution by default for numerical features. Discretizing the features beforehand may increase the performance and accuracy of the algorithm slightly (Kaya, 2008). However, despite the fact that there are optimizations possible, the general performance of Naïve Bayes is highly biased and seems not fit for this classification task.

6.3.2. *Tuning SVM*

SVM can be tuned using two parameters and by selecting different kernels. The γ parameter defines the influence a single instance can have on the algorithm and the parameter C defines the degree of accepting

instances that are not separable. When γ is high, the kernel will look for values in its vicinity and this may result in overfitting. While as the γ becomes low, the reach of the kernel may become too large and the SVM is not able to capture the complexity of the data and will yield similar results to a linear SVM. Since γ is only used in kernels and needs to be low for the kernels to show different behavior compared to linear models, the γ value will be set to $1e^{-12}$. This value will guarantee low bias but may result in high variance. When C is low, it will accept most instances that lie on the wrong side of the hyperplane and when C→ ∞, no instances in the wrong side of the hyperplane are accepted. The C value is more relevant for optimization. There is a wide choice of kernels which can be chosen, but the most popular are the linear SVM and the Gaussian Radian Based Function (RBF) kernel. These functions will therefore be tested and the accuracy of these will be evaluated.

For a linear kernel, a change in the C value has little to no significant effect as the SVM algorithm shows extreme bias. When C=0.1, the sensitivity or specificity is either 0 or 1, indicating that all the keystrokes are either classified as deceptive or truthful. Raising the parameter to 10 does not yield different classification rates. The confusion matrix can be found in Table 12. For all participants, all keystrokes were either classified as deceptive or truthful.

For the RBF kernel, the results were similar to the linear kernel. The C value seemed of no influence on the classification and for all participants, the classification was biased. Meaning that both messages were either classified as truthful or deceptive with a specificity and/or sensitivity being exactly one.

		Classification	
Actual Class		17	13
		17	13

		Classification	
Actual Class		17	13
		17	13

Table 12 - Confusion matrix for the SVM linear (left) and RBF (right) kernel using the feature set

6.3.3. Tuning k-NN

k-Nearest Neighbor algorithm has two parameters that can be used for tuning. The first one is obviously the number k, which should be uneven as an even number may cause ambiguity as the classes are equally present. The second parameter is the distance weighing. For this classification, the popular weighing scheme $\frac{1}{distance}$ is used. This weighing scheme decreases the importance of a neighboring class depending on the distance of the neighbor to the instance that needs to be classified.

The specificity and sensitivity were both distributed around 0.5 for low values of k. The model became more biased with an increase in k, resulting in an increase in the difference between the specificity and sensitivity became larger. The best classification results were the best for k = 3, resulting in the classification performance as shown in Table 13. For 2 participants, both messages were classified incorrectly. For another 4 participants, both messages were classified correctly. For the remaining 23 participants, only one message was classified correctly.

		Classification	
Actual Class		16	14
		13	17

Table 13 - Confusion matrix for kNN classification using the feature set

6.3.4. C4.5

Generally, decision trees may have difficulties with large feature sets. Since there are only a small number of features for each participant, this becomes no problem for the decision tree. Another downside to decision trees is that the trees may grow too deep and overfitting will occur. In order to account for this, pruning nodes may prevent suboptimal splits to occur. In WEKA, default pruning is applied to nodes that only classify two instances to maintain generalizability. The most important factor that WEKA can optimize is the confidence factor *c* that is used for pruning where smaller values incur more pruning.

Tuning the parameter c did not seem to have any significant effect on the results. The specificity and sensitivity were either 0 or 1 for most users, indicating a large classification bias. The confusion matrix with the classification performance is shown in Table 14. For one participant both the messages were classified correctly. For the remaining 29 participants, both messages were either classified as deceptive or truthful.

	Classification	
Actual Class	18	12
	17	13

Table 14 - Confusion matrix for C4.5 classification using the feature set

6.3.5. Random forest

The implementation of random forest is straightforward in WEKA and not many parameters are available for optimization. The max number of features to use per tree does not increase the classification, as there are already not many features. The number of trees used is automatically optimized in WEKA. In Table 15, the confusion matrix of the classification can be found. For one participant, both messages were classified incorrectly. For another 6 participants, both the messages were classified correctly. For the remaining 23 participants, both messages were either classified as deceptive or truthful.

	Classification	
Actual Class	19	11
	15	15

Table 15 - Confusion matrix for RF classification using the feature set

6.4. Classification Banerjee et al. dataset using machine learning

The Banerjee et al. dataset can be classified using machine learning as a benchmark. Not all algorithms will be used as a comparison. The algorithms Naïve Bayes, k-NN and C4.5 are chosen as the classification results were good compared to the other algorithms, the algorithms are well known and are inexpensive. First the algorithms will be tuned, after which the best classification results will be considered.

6.4.1. Naive Bayes

The results for the NB classification are displayed in Table 16. There was exactly one participant for who both messages were classified correctly. For the rest of the participants, both messages were either classified as deceptive or truthful. The classification shows a strong biasedness towards deceptive messages.

	Classification	
Actual Class	64	36
	63	37

Table 16 - Confusion matrix for classification for NB using the feature set

6.4.2. k-NN

The results for k-NN were convincingly the best when k = 9. The results of the classification can be found in Table 17. The classifications shows a strong bias towards deceptive messages. For 2 participants, both the message parts were classified incorrectly. For another 15 participants, both the message parts were classified correctly. For the remaining 83 participants, the parts of both messages were either classified as truthful or deceptive.

	Classification	
Actual Class	77	23
	65	35

Table 17 - Confusion matrix for classification for k-NN using the feature set

6.4.3. C4.5

Lastly, the decision tree C4.5 will be tested. Changing the parameter c did not improve or decrease the classification. A c-value of 0.25 is used for the classification, which is displayed in Table 18. For one participant, the parts of both the messages were classified correctly. For the other participants, the parts of both messages were either classified as deceptive or truthful.

	Classification	
Actual Class	73	27
	72	28

Table 18 - Confusion matrix for classification for C4.5 using the feature set

7. Results

In this chapter, the results of the (exploratory) data analysis phase will be discussed. In sub-chapter 7.1, the two datasets used in this thesis are compared. In sub-chapter 7.2, the results of the distance based classification of both datasets is considered. In sub-chapter 7.3, the results of the machine learning methods of both datasets is considered.

7.1. Dataset comparison

In order to detect changes in the data for both the PwC and Banerjee et al. dataset, statistical tests were done in chapter 5. Each feature of the whole dataset was tested, where all the instances of a feature from one or more deceptive messages were compared to all the instances of a feature from one or more truthful messages. The statistical test was done with the Mann-Whitney U-test. The number of participants for which each feature was found to be significantly different, are displayed in Table 19. The distribution of statistically significant features follows more or less the same pattern for both datasets. Some of the low acceptance rates may be explained due to high variability in the dataset, as the Up – Down and Up – Up flight times may take the highest range of values of all features.

	PwC dataset (n = 30)	Banerjee et al. (n = 100)
Dwell time	9 (30%)	34 (34%)
Down – Down	7 (23%)	29 (29 %)
Down – Up	9 (30%)	31 (31%)
Up – Down	3 (1%)	23 (23%)
Up – Up	3 (1%)	19 (19%)
Typing speed	13 (43%)	53 (53%)
Pause rate	1 (3%)	14 (14%)

Table 19 - Number of participants for which the differences of the dataset were statistically significant

After the statistical tests were done for all participants and features of the whole dataset, the features for each participant were selected. The feature selection process was done using a best-of-three approach. First the difference value between the PDFs of the training set (containing a truthful and deceptive message/part of message) were calculated. The three features with the highest difference value were selected to populate the feature set of that participant. For the pause rate, no features were selected as this feature was modeled differently and classified separately. In Table 20, the adoption rate is presented. The adoption rate is the number of participants that accepted the feature in their feature set (one of the three features with highest difference value). The adoption rate shows similarities between the two datasets, as the Down - Down, Down - Up and Up - Up flight times are very near. It is also remarkable to see that the dwell time was not accepted in both cases.

	PwC dataset (n = 30)	Banerjee et al. (n = 100)
Dwell time	0 (0%)	0 (0%)
Down – Down	18 (60%)	63 (63%)
Down – Up	24 (80%)	82 (82%)
Up – Down	19 (63%)	46 (46%)
Up – Up	25 (83%)	84 (84%)
Typing speed	4 (13%)	7 (7%)

Table 20 - Adoption rate of the features for both datasets

7.2. Distance based classification results

The distance based classification (using a Manhattan metric) was done for both the PwC dataset as well as for the sample of the Banerjee et al. dataset. The classification was also done for the feature set (best of three features) and the pause rate. In Table 21, the results of the classification is displayed. The rows indicate if (parts of the) messages of a participant are both classified correctly (both correct), classified incorrectly (both incorrect) or if the model is biased towards classifying both messages as one type (biased). The distribution of the results are similar. Most of the participants had a classification bias.

	PwC dataset (n = 30)			Banerjee et al. (n = 100)		
	Both correct	**Both wrong**	**One wrong**	**Both correct**	**Both wrong**	**One wrong**
Feature set	1 (3%)	1 (3%)	28 (93%)	1 (1%)	5 (5%)	94 (94%)
Pause rate	3 (1%)	2 (6%)	25 (83%)	2 (2%)	0 (0%)	98 (98%)

Table 21 - Distance based classification results

7.3. Performance of the algorithms

In Table 22, the results for the classification using machine learning algorithms is presented. The rows indicate if (parts of the) messages of a participant are both classified correctly (both correct), classified incorrectly (both incorrect) or if only one message is classified correctly indicating that the data is unclear and not separable. Generally, the classification performance is bad as for most participants the model becomes biased. The performance was the best for RF and k-NN, which had the highest number of participants for which both messages were classified correctly. The results show that the algorithms treat both the datasets in the same manner, which yields equally bad results.

	PwC dataset (n = 30)			Banerjee et al. (n = 100)		
	Both correct	**Both wrong**	**One wrong**	**Both correct**	**Both wrong**	**One wrong**
NB	0 (0%)	0 (0%)	30 (100%)	1 (1%)	0 (0%)	99 (99%)
SVM	0 (0%)	0 (0%)	30 (100%)	-	-	-
k-NN	4 (13%)	2 (6%)	26 (86%)	15 (15%)	2 (2%)	83 (83%)
C4.5	1 (3%)	0 (0%)	29 (96%)	1 (1%)	0 (0%)	99 (99%)
RF	6 (20%)	1 (3%)	23 (76%)	-	-	-

Table 22 - Number of participants for each classification result

8. Conclusion and Discussion

In this chapter, the research questions are answered in sub-chapter 8.1. Sub-chapter 8.2 contains a discussion on the results and sub-chapter 8.3 describes future research directions.

8.1. Conclusion

In this sub-chapter, the (sub) research questions are answered.

S1. What insights does the literature about deception detection yield?

A broad overview of the state of the art of deception detection is given in literature study in paragraph 2.1. The research on deception detection spans decades, focusing on a challenge that has kept humans concerned for millennia. Hypothesis that stem from research indicate that deceiving is a broad term for sending a message to fosters a false believe. In order to be deceptive, the sender has to write a message that contains a statement with which the sender intrinsically disagrees or knows is not (entirely) true. Since real world deceptive detailed keystroke data is hard to conceive, an experiment had to be designed to force participants into being deceptive. In this experiment, participants were asked their opinion about a subject they cared about: their work. The questions were designed in collaboration with an experienced senior manager at PwC. It was mentioned that the research was anonymous in order for people to feel safe to answer the questions. The experiment was also designed to be full disclosure. The experiment deviated from an existing experiment by gathering more questions (a total of 4 messages) and by using different questions throughout the experiment such that for each task, a new answer had to be generated. Once the opinions of four statements were answered, two opinions were randomly selected and reversed. The participant was then asked to defend the four statements, two statements to explain their true opinion and two statements that they disagreed with. This design was created using literature (Carlson et al., 2004) to create the experiment in a CMC-environment.

The first insight from the literature on deception detection is a direct consequence of the definition of a deceptive message (Buller & Burgoon, 1996). This definition states that a deceptive message can take a lot of forms ranging from a full fabricated lie, to a half-truth, concealment and vagueness. Due to this broad definition, a scope on the interpretation of deception has to be established to prevent ambiguity in the research design. Other important insights come from the four-factor (Zuckerman et al., 1981) and the leakage hypothesis (Ekman & Friesen, 1969). These theories build upon each other where the four-factor theory defines factors that the deceiver experiences when deceiving. Arousal is defined to be the most prominent factor. The leakage hypothesis describes how these factors manifest itself in a behavioral and physical manner. More recent studies and surveys describe in more detail how the people expect deceivers to behave and how deceivers actually behave (Vrij, 2008). People tend to be very bad in deception detection as the accuracy of detecting deception is often equal to the chance of guessing or worse due to biasedness (Bond & DePaulo, 2006). Research on deceiving in a CNC environment is scarce.

S2. What characteristics can be extracted from the typing rhythm data that change when people are deceptive and when they are not?

In paragraph 2.2, an extensive study was done on the chronology of keystroke dynamics. The literature overview covers the methods and characteristics that were used to assess topics like authentication, identification and emotion recognition. Some scholars mention that the challenge in keystroke dynamics is finding new characteristics (or features). However, some features already have so much success in the earlier mentioned topics that they are referred to as behavior biometrics. These were treated in Chapter 4. This thesis asserted that given the fact that other (behavioral) biometrics change when people are deceptive, keystroke dynamics such as a biometric may also change. The features that can be considered a biometric are dwell time (the time a person holds a key), the flight time (the time it takes to press and/or release consecutive keys, the typing speed, the capitalization rate, the special key rate, the deletion rate, the pause rate, punctuation rate, unrelated keys rate and special key rate. Now these features have a lot of overlap. Some features were identical or not suitable for this thesis. Therefore, the most important behavioral features were chosen to test the hypothesis. These features were the dwell time, digraph flight time and the typing speed. Also the pause rate between words was considered separately.

In Chapter 5, an exploratory study was done to enhance the intuition about the data and to discover possible new patterns. Some statistical tests were performed on the dataset next to a feature selection phase, in which a

difference value was calculated. There was a large difference between the results of the statistical tests and the difference values. This might be due to the standard deviation of some of the features, which resulted in a decrease in statistical significance for the users. The dwell time was eventually not chosen at all. The dwell time was chosen in 95% of the cases. The typing speed approximately a third of the cases. Other (quantitative) characteristics were also studied in this thesis. The quantitative characteristics analyzed were the lengths of the messages in time and key events, and the deletion rate. These characteristics did not seem to be distinguishable visually and where not used for analysis.

S3. What are the methods that one can use to discriminate between truthful and deceptive messages using only the keystroke data?

Since the accessibility of fast computers and the possibility to store and exchange data, machine learning has taken a flight in the last decades. In chapter 2, the chosen algorithms are chosen based on their performance and popularity. Since no baseline classification on deceptive messages has been performed yet, the performance of these methods is assessed. Each message can be modeled as a collection of key events described by biometrical keystroke dynamics features. Each message therefore becomes a dataset of instances (i.e. key events) and different machine learning methods can be used to classify or discriminate truthful and deceptive messages. The machine learning methods were then optimized by using performance measures defined in paragraph 2.4. The classification was done for each participant individually, as keystroke dynamics is a biometric has different classification requirements for each participant. The algorithms were Naive Bayes, Support Vector Machine, k-Nearest Neighbor, C4.5 and Random Forest. Next to machine learning, a distance metric classification was also applied as this has been effectively applied in the past. The best performing algorithms were then chosen to assess the predictability of the messages of each of the user and the final results in chapter 7 showed the performance of the algorithms on the datasets.

S4. How does the dataset in this thesis compare to the dataset from the other researchers?

There is a corpus available that was collected by Banerjee et al. The PwC dataset was compared to a random sample (n = 100) out of the dataset of Banerjee et al. The corpus has the disadvantage that there is only one type of message available per participant. That means there is one truthful and one deceptive message (essay) available for each unique individual. The questions for each participant were also the same. This can be a disadvantage, because being deceptive can become equal to negating the previous statement. In their paper, each message was modelled as an instance and the quantitative properties were used for classification, yielding a high classification rate (with stylometry). The PwC dataset has 4 messages, from which two deceptive and two truthful. The truthful and deceptive assignments were randomized and each question was different.

Despite the differences, it was found that the classification results, statistical significance, feature selection were equally distributed for both datasets. This might indicate that both datasets have a similar quality and that there is no discrimination between the two samples for this assignments.

RQ: Can keystroke dynamics be used to detect deceptive messages without looking at the contents of the message?

In the results of chapter 7, it is found that machine learning methods perform equally bad or worse than guessing. The literature on deception detection stated that humans are as bad as chance, most of the time even worse than guessing due to biasedness. The same performance is found when classifying deceptive messages with the PwC dataset. The data does not seem separable in most cases as all the keystrokes of both the truthful and deceptive message are classified as if they would belong to either one class. The methods could often not discriminate the two types of messages. It can therefore be stated that with the given experiment and dataset, machine learning is not better in predicting deceptive messages than a human would be able to guess. The comparison with the Banerjee et al. dataset seems to confirm this. Using these features, this dataset, this experiment and these methods, correct classification is not possible.

8.2. Discussion

In most studies on deception, researchers instructed their participants to lie during experiments. Another common way to acquire data to study deception was done by studying tapes or recordings of courtrooms and interrogations. In other words, places where truth and lie eventually untangle and where the nonverbal

behavioral clues can be studied afterwards. Although some studies show some success with these procedures, it is not necessarily true that instructed deception does not tamper with the arousal, negative effect or discomfort that genuine deception induces.

This study instructs users to lie, as most studies do. A weakness in this design may be that an instruction to deceive can be considered a form of obedience. It is possible that obedience can tamper with the hypothesized effects of deception mentioned in the literature. These effects are important because they are said to be the driving force behind the leakage of non-verbal clues as stated by the leakage-hypothesis (Ekman & Friesen, 1969). Moreover, voluntary deception is more common in practice than instructed deception. The participant could feel less guilty about following morally doubtful instructions, as shown by the famous Milgram Experiment (Milgram, 1963). However, this weakness can be addressed with two counterarguments.

First of all, it is very hard to create an experiment that complies to the non-instructive criteria without involuntarily creating other questionable causal variables. Imagine an experiment where one participant has to deceive another participant. Then this experiment will only work on the premise that there is an (assumed) information asymmetry between the sending and the receiving party. Otherwise no false belief can be fostered. There must be an incentive that will motivate the participant to make up a deceptive message, as people are usually guilt averse (López-Pérez & Spiegelman, 2013). The deceptive message has to be quite long because enough keystrokes have to be recorded for analysis. In order for a participant to be able to consciously depart from some contextual truth, a lot of contextual truth has to be created. This way the game becomes information intensive. It also means that if the receiver decides to depart from this contextual truth, he has to make up his own deceptive context. But does that mean that making up a story equals deceiving? And if it does, why are people not instructed to make up a story nonetheless? How can it be measured, and to what degree do participants want to be credulous? This results in many philosophical questions about deception and the human emotions that go beyond the scope of this thesis, but may be eligible for further research.

Secondly, it is not known how people experience their own deceptive behavior in a CMC-environment. It seems obvious to use the four-factor theory (Zuckerman et al., 1981) from real life deception to deception in a CMC-environment. However, this is not necessarily true. Deception in a CMC-environment is different as visible behavioral clues are not apparent. This reduces the chance to be caught and might influence the arousal. Also, the deceiver is often alone, can think more about his answers and does not directly face the consequences of his lying. Furthermore, we found no research on how deceivers feel after or while lying in a CMC-environment. Therefore, these factors should not and cannot be extrapolated in this research.

Evaluations with some participants after the study revealed that most participants were aware that they were expected to defended a statement with which they did not agree. In some cases, the participant did have the idea that they behaved differently. The participants indicated that they thought differently when they had to write a deceptive statement. Since each participant categorized their own typing skill as average or good, it is possible that due to the proficiency in typing, the biometrical behavior is influenced or mitigated. Thereby, leading to the creation of a homogenous set where detection deception becomes very diligent and difficult. It may also be the case that no deception can be detected through analyzing keystrokes, but to confirm this more research has to be done.

8.3. Further research

In order to gain more insight, better datasets could be supplied. Although it is hard to find a good source for data collection, putting participants under more pressure could yield different outcomes. The downside to this is that the research may become unethical. Another direction is that more data should be collected. More keystrokes can indicate a better convergence towards the true values of the features. In order to do so and not to fatigue the participants, the data collection phase can be extended over the course of multiple days. By following the behavior of multiple participants over a longer period of time, more qualitative study can be done to aberrant behavior and new insights might generate more insights on a more generic approach to the research.

9. Bibliography

Banerjee, R., Feng, S., Kang, J. S., & Choi, Y. (2015). *Keystroke Patterns as Prosody in Digital Writings : A Case Study with Deceptive Reviews and Essays*.

Bond, C. F., & DePaulo, B. M. (2006). Accuracy of deception judgments. *Personality and Social Psychology Review : An Official Journal of the Society for Personality and Social Psychology, Inc*, *10*(3), 214–234. https://doi.org/10.1207/s15327957pspr1003_2

Bours, P. (2012). Continuous keystroke dynamics: A different perspective towards biometric evaluation. *Information Security Technical Report*, *17*(1–2), 36–43. https://doi.org/10.1016/j.istr.2012.02.001

Bours, P., & Mondal, S. (2014). Continuous Authentication using Fuzzy Logic, (SEPTEMBER). https://doi.org/10.1145/2659651.2659720

Buller, D. B., & Burgoon, J. K. (1996). Interpersonal Deception Theory. *J. Seiter & R. Gass (Eds.), Readings in Persuasion, Social Influence, and Compliance Gaining*, 46.

Carlson, J., George, J., Burgoon, J., Adkins, M., & White, C. (2004). Deception in Computer-Mediated Communication. *Group Decision and Negotiation*, *13*, 5–28. https://doi.org/10.1023/B:GRUP.0000011942.31158.d8

Chapman, J. L. (2012). Criminalistics and Court Expertise 2012 Annual Issue , Number 57, (57), 238–251.

De Ru, W. G., & Eloff, J. H. P. (1997). Enhanced password authentication through fuzzy logic. *IEEE Expert-Intelligent Systems and Their Applications*, *12*(6), 38–45. https://doi.org/10.1109/64.642960

DePaulo, B. M., Lindsay, J. J., Malone, B. E., Muhlenbruck, L., Charlton, K., & Cooper, H. (2003). Cues to deception. *Psychological Bulletin*, *129*(1), 74–118. https://doi.org/10.1037//0033-2909.129.1.74

DePaulo, Lanier, & Davis. (1983). DePaulo_Lanier_Davis_1983.pdf.

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, *55*(10), 78. https://doi.org/10.1145/2347736.2347755

Dowland, P. S., Fvrnell, S. M., & Papadaki, M. (2002). KEYSTROKE ANALYSIS AS A METHOD OF ADVANCED USER AUTHENTICATION AND RESPONSE.

Ekman, P., & Friesen, W. (1969). The repertoire of nonverbal behavior: categories, origins, usage and coding.

Ekman, P., O'Sullivan, M., & Frank, M. G. (1999). A-Few-Can-Catch-A-Liar-Psychological-Science.pdf. Psychological Science.

Elkins, A. C., Zalfeiriou, S., Burgoon, J., & Pantic, M. (2014). *Unobtrusive Deception Detection*. *Handbook of Affective Computing, R. Calvo, S. DMello, A. Kappas and J. Gratch Eds.. Springer*.

Epp, C., Lippold, M., & Mandryk, R. L. (2011). Identifying emotional states using keystroke dynamics. *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems*, 715–724. https://doi.org/10.1145/1978942.1979046

Gaines, R. S., Lisowski, W., James Press, S., & Shapiro, N. (1980). Authentication by keystroke timing: some preliminary results. National science foundation.

George, J. F., Marett, K., Burgoon, J. K., Crews, J., Cao, J., & Lin, M. (2004). TRAINING TO DETECT DECEPTION : AN EXPERIMENTAL INVESTIGATION, *0*(C), 1–10.

Giot, R., El-abed, M., & Rosenberger, C. (2009). Keystroke Dynamics Authentication For Collaborative Systems.

Gould, S. (2005). A Novel Approach to User Authentication Through Machine Learning of Keyboard Acoustic Emanations, 1–5.

Grazioli, S. (2004). Where Did They Go Wrong ? An Analysis of the Failure of Knowledgeable Internet Consumers to Detect Deception Over the Internet. *Kluwer Academic Publishers*, 149–172.

Grazioli, S., & Wang, A. (2001). Looking Without Seeing : Understanding Unsophisticated Consumers ' Success and Failure To Detect Internet Deception. *International Conference on Information Systems (ICIS)*, 193–204.

Hosseinzadeh, D., & Krishnan, S. (2008). Gaussian mixture modeling of keystroke patterns for biometric applicatoins. *IEEE Transactions on Systems*, *38*(6), 816–826.

Kaya, F. (2008). Discretizing Continuous Features for Naive Bayes and C4.5 Classifiers. *University of Maryland Publications*. Retrieved from http://cgis.cs.umd.edu/Grad/scholarlypapers/papers/fatih-kaya.pdf

Killourhy, K. S., & Maxion, R. A. (2009). Comparing Anomaly-Detection Algorithms for Keystroke Dynamics.

Kolakowska, A. (2010). Generating training data for SART-2 keystroke analysis module. *Proceedings of the 2nd International Conference on Information Technology (ICIT '10)*, 57–60.

Lau, E., Liu, X., Xiao, C., & Yu, X. (2004). Enhanced User Authentication Through Keystroke Biometrics.

Leal, S., Vrij, A., Warmelink, L., Vernham, Z., & Fisher, R. P. (2013). You cannot hide your telephone lies: Providing a model statement as an aid to detect deception in insurance telephone calls. *The British Psychological Society*, 18.

Lim, Y. M., Ayesh, A., & Stacey, M. (2014). Detecting Cognitive Stress from Keyboard and Mouse Dynamics during Mental Arithmetic. *Science and Information Conference 2014*, 146--152.

López-Pérez, R., & Spiegelman, E. (2013). Why do people tell the truth? Experimental evidence for pure lie aversion. *Experimental Economics*, *16*(3), 233–247. https://doi.org/10.1007/s10683-012-9324-x

Milgram, S. (1963). Behavioral Study of Obedience. *Journal of Abnormal Psychology*, *67*(4), 371–378. https://doi.org/10.1037/h0040525

Monrose, F., & Rubin, A. D. (1997). Authentication via Keystroke Dynamics.

Monrose, F., & Rubin, A. D. (2000). Keystroke dynamics as a biometric for authentication, *16*, 351–359.

Moskovitch, R., Feher, C., Messerman, A., Kirschnick, N., Mustafić, T., Camtepe, A., … Elovici, Y. (2009). Identity theft, computers and behavioral biometrics. *2009 IEEE International Conference on Intelligence and Security Informatics, ISI 2009*, 155–160. https://doi.org/10.1109/ISI.2009.5137288

Nguyen, T. T., Le, T. H., & Le, B. H. (2010). Keystroke dynamics extraction by independent component analysis and bio-matrix for user authentication. *Proceedings of the 11th Pacific Rim Internatoinal Conference on Trends in Artificial Intelligence*, 477–486.

Patil, V. P., Nayak, K. K., & Saxena, M. (2013). Voice Stress Detection. *International Journal of Electrical, Electronics and Computer Engineering*, *2*(2), 148–154. Retrieved from http://researchtrend.net/ijet21/ijetnew/24 VIJAY PATIL.pdf

Revett, K., Gorunescu, F., Gorunescu, M., Ene, M., & Santos, H. M. D. (2007). A machine learning approach to keystroke dynamics based user authentication Sérgio Tenreiro de Magalhães and, *1*(1).

Revett, K., Magalhães, S. T. De, & Santos, H. M. D. (2005). -Enhancing Login Security Through the Use of

Keystroke Input Dynamics, 661–667. Retrieved from http://link.springer.com/chapter/10.1007/11608288_88

Rybnik, M., Panasiuk, P., Saeed, K., & Rogowski, M. (2012). Advances in the Keystroke Dynamics : the Practical Impact of Database Quality.

Shmueli, G., & Koppius, O. R. (2011). Predictive analytics in Information Systems research. *MIS Quarterly*, *35*(3), 553–572.

Teh, P. S., Teoh, A. J., & Yue, S. (2013). A Survey of Keystroke Dynamics Biometrics, *2013*.

Vizer, L. M., Zhou, L., & Sears, A. (2009). Automated stress detection using keystroke and linguistic features: An exploratory study. *International Journal of Human Computer Studies*, *67*(10), 870–886. https://doi.org/10.1016/j.ijhcs.2009.07.005

Vrij, A. (2008). *Detecting lies and deceit: pitfalls and opportunities. Wiley Series in the Psychology of Crime, Policing and Law*. *Analysis*. https://doi.org/10.1002/1521-3773(20010316)40:6<9823::AID-ANIE9823>3.3.CO;2-C

Wang, J. T., Spezio, M., Camerer, C. F., The, S., Economic, A., June, N., … Camerer, C. F. (2010). American Economic Association Pinocchio â€™ s Pupil : Using Eyetracking and Pupil Dilation to Understand Truth Telling and Deception in Sender-Receiver Games Published by : American Economic Association Stable URL : http://www.jstor.org/stable/27871237 Pino, *100*(3), 984–1007.

Wu, X., Kumar, V., Ross, Q. J., Ghosh, J., Yang, Q., Motoda, H., … Steinberg, D. (2008). *Top 10 algorithms in data mining*. *Knowledge and Information Systems* (Vol. 14). https://doi.org/10.1007/s10115-007-0114-2

Xiaojun, C., Zicheng, W., Yiguo, P., & Jinqiao, S. (2013). A Continuous Re-Authentication Approach Using Ensemble Learning. *Procedia Computer Science*, *17*, 870–878. https://doi.org/10.1016/j.procs.2013.05.111

Zhong, Y., & Deng, Y. (2015). CHAPTER 1 A Survey on Keystroke Dynamics Biometrics: Approaches, Advances, and Evaluations. *Recent Advances In User Authentication Using Keystroke Dynamics Biometrics*, *2*, 1–22. https://doi.org/10.15579/gcsr.vol2.ch1

Zuckerman, M., DePaulo, B. M., & Rosenthal, R. (1981). Verbal and non-verbal communication of deception. In L. Berkowitz (ed.). *Advances in Experimental Societal Psychology*, *14*, 1–59.

A. Experimental design

A.1. Instrumentation

A.1.1. Web environment

In order to be able to reach out to the whole business unit, an accessible environment had to be created. Key log functionality is readily available in most programming languages, so choosing a platform is strongly dependent on the needs of the study. In this paragraph it is described what environment is chosen, how and in what format the keys will be logged and the reliability of the key logger is assessed.

In order to reach out to the whole business unit, some conditions need to be considered. The target group consists out of employees (consultants) who are often on the road. The data collection took place during the holidays, so the office was quite empty. Due to strict policies it was not convenient to distribute software throughout the company. In order to be able to reach all the employees of the business unit, it was decided that the appropriate environment for this experiment would be a web environment.

The website was hosted on a webserver on the domain abhuisman.nl. The webpage was built with the framework Bootstrap (v. 3.3.6) which depends on a stripped version of jQuery (v. 1.12.3.min). The webpage needs to be able to log the keystrokes on the client-side and must then pass the keystrokes on to the server-side. This way the log files can be stored on the server.

The group of participants use a corporate laptop. This laptop has a 64-bit Windows 8.1 Enterprise edition installed. The laptop is a Lenovo ThinkPad (i7-4600U @ 2.1 GHz, 12 GB RAM) and the updates on the laptop are done automatically. This means that most laptops run the standard browser Internet Explorer 11 (v.11.0.31) at the time of writing. However some may use other browsers. The website should work on the common browsers (Chrome, IE, Firefox, Safari and Opera). The bootstrap framework supports all these browsers mostly with backwards compatibility with the exception of some old versions of Internet Explorer. But since the updates are done automatically, it is expected that most users will have the newest version of IE.

A.1.2. JavaScript Key logger

This thesis will focus on collecting the timestamp of keystrokes. A timestamp with an accuracy of 1 millisecond is deemed sufficient (Hosseinzadeh & Krishnan, 2008). Other attempts as a source of features have been tested, like keystroke pressure (Revett et al., 2005), typing sequence difficulty (De Ru & Eloff, 1997), frequency of typing error (Kolakowska, 2010) and sound of typing (Nguyen, Le, & Le, 2010). But these acquisition methods are in general not very popular. They are also hard to acquire and have no practical added value to the study yet, as the techniques are often not applied.

The key logger will be written in JavaScript, which is the most popular client-side programming language available for web browsers. There is a library for JavaScript called jQuery that automated and optimized a lot of JavaScript functionality. jQuery has three functions available for the logging of keys. The `keypress`, `keydown` and `keyup` event. According to the documentation[7], the keypress event is sent to an element when the browser registers keyboard input. The event is similar to the `keydown` event, but also registers when a key stays pressed. It triggers each time another character is inserted. The `keydown` and `keyup` event trigger each time a key is pressed or released respectively. The events can be attached to any element in the browser. Since we want to log continuously over the course of the experiment, the events will be attached to the `document` object. This is done because all keypresses will eventually find their way up the DOM to the `document` element, due to event bubbling. The only way keys are logged, is when the focus is on the browser.

When the keyup and keydown functions are triggered, they return an event with information on what happened. The event contains some useful information, for example if the alt/control/shift key was pressed simultaneously, a timestamp, a key/char code together with the corresponding key, and the type of the event (key up or down). This data can be thankfully used to store important aspects on the keystrokes.

The event contains a timestamp. Naturally, JavaScript contains certain methods to accurately log the time. It is possible to use the function `Date.now()` logs time in milliseconds. One can use the stopwatch object which can log ticks at a certain moment to a high degree of accuracy. One can also use `performance.now()` to get a high

[7] http://api.jquery.com/{keypress, keyup, keydown}/

degree of accuracy. Initially `Date.now()` was used. This has the disadvantages that it is executed after the event was sent. So when there are some other intensive tasks running on the client side (or on the laptop in general), lags could occur. A few tests showed that the value in `event.timeStamp` always had a lower value compared to the other two functions. Therefore the timestamp in the event is considered most accurate and is used as the timestamp. This has the advantage that if the event is not yet handled, the user will also receive no feedback on his screen. And might adjust his behavior accordingly. The timestamp was rounded to the smallest integer, resulting in an integer in milliseconds.

The event sent by the functions also contained either a char code or a key code. This depended on the browser and caused some confusion. jQuery however, handles it nicely. It normalizes the event to the `event.which` property which contains the true code for the button pressed.

In order to prevent some unwanted behavior, some functionality in the web environment is cancelled. The tab key is cancelled as it yields no added value and might cause some unwanted behavior. The control button is also disabled, as it is not desirable that participants will copy pieces of text. In that order, the alt key was also disabled to prevent people from pressing alt-tab to move to another window during the experiment.

Now that the keystrokes are registered with an accompanying timestamp, the logs need to be sent to the server. JavaScript can send GET and POST requests to servers with the `XmlHttpRequest` object. In order to send a request, an instance of the object has to be created. The instance requires the specification of the request (GET or POST). It also needs the URL to which the request should be sent. Then some headers need to be sent in order for the server to understand what is being received. And finally, the request can be send accompanied by the parameters that contain the information. The time the request takes is not very important for the quality of the data, since the event generated at the key interactions contains the parameters. In case it happens that one requests arrives earlier at the server than a request that was created earlier, a counter also sends a variable with the request. This is done such that the order in which the requests are sent (and consequently, the order in which the events were triggered) can be backtracked if any delays occur.

Only IE is able to distinguish left from right control-, alt- and shift keys. [8] Shift keys are said to be an important distinction feature for identification. For deception, this might not be the case. It does not sound rational that participants will start using the right shift key instead of the left one when they are deceptive.

The code for the JavaScript key logger can be found in Appendix A.1 together with the PHP code in Appendix A.3 to handle the AJAX requests.

A.1.3. Log format

The format in which the keystrokes will be logged is stored in a csv-format. CSV stands for comma separated values and is easy to process. The logged format is shown in Table 23.

	COUNTER	**KEYCODE**	**TIMESTAMP**	**TYPE**
Type:	Integer $\in \mathbb{N}$	Integer $\in \mathbb{N} \cap [4,222]$	Integer $\in \mathbb{N}$	Character $\in \{D, U\}$
Example:	215	88	1486422506	D

Table 23 - CSV format for the logged keystrokes

The counter keeps track of the sequence as is processed by the browser. The counter is incremented each time an event (keyup or keydown) is triggered. The key code is retrieved from the `event.which` property and resembles ACII code in most cases. There are some exceptions depending on the older browser versions, so ideally the browser version should also be logged. The timestamp is extracted from the event, as is mentioned earlier. The type stands for either a keydown ('D') or a keyup ('U') event.

While analyzing every possible key event, it was found that not all keys were used during research. From scanning all the messages, the results were that there are 59 distinct charcodes that can be used as nominal values for analysis.

[8] http://help.dottoro.com/ljgjxtkf.php

A deficiency with requests from the client to the server side (and as often happens with communication protocols) is that sometimes the request do not reach the server. While some trial results were analyzed, it was seen that this is also the case for the `XmlHttpRequests`. This can have many causes. Either the client is too busy with (too many) other tasks, or the server is temporarily unavailable. In order get a good idea of what the webserver is able to handle, a small script is written in JavaScript to simulate human typing behavior and assess the reliability of the webserver.

The typing speed depends largely on the typing skills of the participant. A small assessment showed that the time between the keyup and –down events is usually not smaller than 50 milliseconds. Also, the time between two consecutive keydown events is about 200 milliseconds. The aim of the study is to collect at least 2000 key events resulting in 4000 keyup and -down events. The reliability can be assessed with the use of the functions `setInterval()` and `setTimeout()` to loop a keyup and –down event consecutively. The time between the keyup and –down is set to 50 milliseconds. The time between a new key event is set to 200 milliseconds. Over the course of half an hour the requests were done continuously, which resulted in about 18000 keypresses logged.

This means that approximately 4 people can type simultaneously. In Table 24, the results for a small simulation are found. The clients ran simultaneously. It is shown that per message there will always be a specific error. This will result in more work during the data handling and a less reliable pattern.

Sent messages	Non arrivals	Successful requests / error	Error probability / request
2952	4	738	0.001355
2574	2	1287	0.000777
1822	3	607	0.001097
3352	2	1676	0.000589

Table 24 - XMLHttpRequest statistics

In order to account for these inaccuracies, the number of simultaneous requests are reduced by sending batches of events to the server instead of two messages per key event. The batch size was set to 10 which means that five key events (one key-up & one key-done) are enough to make a request. This way, the total number of requests to the servers are divided by 10. In Table 25, the statistics for four simultaneous clients are shown.

Sent requests	Non arrivals	Successful requests / error	Error probability / request
1004	3	335	0.002988
2140	0	2140	0
2040	0	2040	0
2110	0	2110	0

Table 25 - XMLHttpRequest batch statistics

Due to the fact that there are a lot less requests, the non-arrivals have gone down while the same number of key events are sent. In order to be really sure that all keystrokes arrive, another measure is built in. A `XMLHttpRequest` that receives another HTTP response other than `200 (OK)`, gets send again within a second. This process is repeated until the request finally arrives. For a final time, four simultaneous clients were initiated that performed key-events on the webpage every 200 milliseconds. All initiated requests arrived correctly and there were no errors.

A.2. Keylogger in JavaScript

```
$(document).on('keyup keydown', function(e){
        //Remove Shift, Ctrl & Alt
        if(x == 9 || x == 18){e.preventDefault();}
        else{
                //Checks if the variable record is set to true
                if(record){
                        var timestamp = Math.round(e.timeStamp);
                        var x = e.which;
                        sender++;
                        count++;
                        e = e || window.event
                        var x = e.charCode || e.keyCode;
                        var eventType;
                        if(e.type == 'keyup'){eventType = 'U';}else{eventType = 'D'};
                        //Create datastring
                        recording += count+","+x+","+timestamp+","+eventType+"\r\n";
                        //If there are 10 items in the recording string, send the log to the server
                        if(sender == 10){
                                //Create XMLHttpRequest
                                var message = recording;
                                var http = new XMLHttpRequest();
                                http.open("POST","keylogger5.php",true);
                                http.setRequestHeader("Content-type","application/x-www-form-
                                urlencoded");

                                //Resends the ajax request if the server times out.
                                http.onreadystatechange=function() {
                                        if (http.readyState === 4){   //if complete
                                                if(http.status === 200){  //check if "OK" (200)
                                                        //success - do nothing
                                                } else {
                                                        //resend
                                                        sendLine(message);
                                                }
                                        }
                                }
                                http.send("log="+encodeURIComponent(message)+"&session="+session);
                                sender = 0;
                                recording = "";
                        }
                }
        }
} );


function sendLine(message){
        var http = new XMLHttpRequest();
        http.open("POST","keylogger5.php",true);
        http.setRequestHeader("Content-type","application/x-www-form-urlencoded");

        //Resend if not complete
        http.onreadystatechange=function() {
                if (http.readyState === 4){   //if complete
                        if(http.status === 200){  //check if "OK" (200)
                                //success - do nothing
                        } else {
                                //resend
                                setTimeout(function(){sendLine(message);}, 1000);
                        }
                }
        }
        http.send("log="+encodeURI(message)+"&session="+session);
}




$(document).on('keyup keydown', function(e){
```

A.3. Receiver in PHP

```php
<?php
$log            =       $_POST['log'];
$session        =       $_POST['session'];
$filename       =       $session . ".txt";
$fp             =       fopen($filename, "a");

fwrite($fp, $log);
fclose($fp);
?>
```

A.4. JavaScript Char Codes

Key	Code
backspace	8
tab	9
enter	13
Shift	16
ctrl	17
alt	18
pause/break	19
caps lock	20
escape	27
page up	33
page down	34
end	35
home	36
left	37
up arrow	38
right arrow	39
down arrow	40
insert	45
delete	46
0	48
1	49
2	50
3	51
4	52
5	53
6	54
7	55
8	56
9	57
a	65
b	66
c	67
d	68

Key	Code
e	69
f	70
g	71
h	72
i	73
j	74
k	75
l	76
m	77
n	78
o	79
p	80
q	81
r	82
s	83
t	84
u	85
v	86
w	87
x	88
y	89
z	90
left window key	91
right window key	92
select key	93
numpad 0	96
numpad 1	97
numpad 2	98
numpad 3	99
numpad 4	100
numpad 5	101
numpad 6	102
numpad 7	103

Key	Code
numpad 8	104
numpad 9	105
multiply	106
add	107
subtract	109
decimal point	110
divide	111
f1	112
f2	113
f3	114
f4	115
f5	116
f6	117
f7	118
f8	119
f9	120
f10	121
f11	122
f12	123
num lock	144
scroll lock	145
semi-colon	186
equal sign	187
comma	188
dash	189
period	190
forward slash	191
grave accent	192
open bracket	219
back slash	220
close braket	221
single quote	222
Space	

as published by https://www.cambiaresearch.com/articles/15/javascript-char-codes-key-codes based on the jQeury normalized `event.which` parameter.

B. Exploratory analysis results

B.1. Frequencies of char codes in dataset

Char code	Deceptive	Truthful	Char code	Deceptive	Truthful	Char code	Deceptive	Truthful
1	0	0	**75**	585	600	**149**	0	0
2	0	0	**76**	789	842	**150**	0	0
3	0	0	**77**	547	515	**151**	0	0
4	0	0	**78**	2322	2345	**152**	0	0
5	0	0	**79**	1303	1374	**153**	0	0
6	0	0	**80**	445	420	**154**	0	0
7	0	0	**81**	2	9	**155**	0	0
8	2736	2907	**82**	1348	1470	**156**	0	0
9	0	0	**83**	840	876	**157**	0	0
10	0	0	**84**	1727	1816	**158**	0	0
11	0	0	**85**	337	335	**159**	0	0
12	0	0	**86**	571	545	**160**	0	0
13	18	4	**87**	528	482	**161**	0	0
14	0	0	**88**	19	35	**162**	0	0
15	0	0	**89**	31	39	**163**	0	0
16	1841	2051	**90**	214	224	**164**	0	0
17	18	61	**91**	1	4	**165**	0	0
18	1	3	**92**	0	0	**166**	0	0
19	0	0	**93**	0	0	**167**	0	0
20	2	2	**94**	0	0	**168**	0	0
21	0	0	**95**	0	0	**169**	0	0
22	0	0	**96**	0	0	**170**	0	0
23	0	0	**97**	0	0	**171**	0	0
24	0	0	**98**	0	0	**172**	0	0
25	0	0	**99**	0	0	**173**	0	0
26	0	0	**100**	0	0	**174**	0	0
27	0	0	**101**	0	0	**175**	0	0
28	0	0	**102**	0	0	**176**	0	0
29	0	0	**103**	0	0	**177**	0	0
30	0	0	**104**	0	0	**178**	0	0
31	0	0	**105**	0	0	**179**	0	0
32	4382	4531	**106**	0	0	**180**	0	0
33	1	0	**107**	0	0	**181**	0	0
34	0	0	**108**	0	0	**182**	0	0
35	11	10	**109**	0	0	**183**	0	0
36	1	5	**110**	0	0	**184**	0	0
37	207	147	**111**	0	0	**185**	0	0
38	7	1	**112**	0	0	**186**	7	6
39	93	58	**113**	0	0	**187**	2	2
40	5	2	**114**	0	0	**188**	103	117
41	0	0	**115**	0	0	**189**	14	5
42	0	0	**116**	0	0	**190**	244	236
43	0	0	**117**	0	0	**191**	10	9
44	0	0	**118**	0	0	**192**	2	0
45	0	0	**119**	0	0	**193**	0	0
46	52	4	**120**	0	0	**194**	0	0
47	0	0	**121**	0	0	**195**	0	0
48	30	23	**122**	0	0	**196**	0	0
49	5	4	**123**	0	0	**197**	0	0
50	2	0	**124**	0	0	**198**	0	0
51	2	3	**125**	0	0	**199**	0	0
52	2	5	**126**	0	0	**200**	0	0
53	1	2	**127**	0	0	**201**	0	0
54	0	1	**128**	0	0	**202**	0	0
55	0	1	**129**	0	0	**203**	0	0
56	1	0	**130**	0	0	**204**	0	0
57	13	20	**131**	0	0	**205**	0	0
58	0	0	**132**	0	0	**206**	0	0
59	0	0	**133**	0	0	**207**	0	0
60	0	0	**134**	0	0	**208**	0	0
61	0	0	**135**	0	0	**209**	0	0
62	0	0	**136**	0	0	**210**	0	0
63	0	0	**137**	0	0	**211**	0	0
64	0	0	**138**	0	0	**212**	0	0
65	1685	1691	**139**	0	0	**213**	0	0
66	313	365	**140**	0	0	**214**	0	0
67	488	423	**141**	0	0	**215**	0	0
68	1162	1206	**142**	0	0	**216**	0	0
69	4351	4421	**143**	0	0	**217**	0	0
70	185	198	**144**	0	0	**218**	0	0
71	550	603	**145**	0	0	**219**	2	2
72	497	470	**146**	0	0	**220**	5	0
73	1584	1653	**147**	0	0	**221**	0	0
74	278	312	**148**	0	0	**222**	39	41
						Total	32561	33536

B.2. Statistical test for Banerjee et al.

	U-value	Dwell time	Down-down	Down-up	Up-down	Up-up	Typing speed	U-value	Pause rate
1	562457,5	0,00	0,11	0,73	0,01	0,14	0,00	17128	0,45
2	2883565	0,32	0,66	0,23	0,84	0,89	0,26	67327	0,03
3	302538,5	0,69	0,74	0,77	0,64	0,52	0,19	8184	0,55
4	1229251,5	0,00	0,85	0,04	0,11	0,99	0,63	27900	0,61
5	325574,5	0,05	0,00	0,00	0,05	0,03	0,00	9715	0,11
6	207839,5	0,85	0,98	0,92	0,94	0,85	0,85	6156	0,85
7	883239	0,37	0,01	0,03	0,05	0,11	0,00	20798	0,27
8	571880	0,71	0,67	0,93	0,87	0,72	0,29	12395	0,57
9	588052	0,65	0,16	0,20	0,11	0,13	0,05	16218	0,81
10	288652	0,36	0,73	0,50	0,94	0,97	0,67	7625	0,98
11	182275	0,68	0,57	0,48	0,59	0,65	0,01	5832	0,52
12	441974	0,19	0,76	0,75	0,92	0,92	0,47	13130	0,32
13	345420	0,89	0,00	0,00	0,00	0,00	0,00	9367	0,27
14	406392	0,39	0,03	0,07	0,04	0,06	0,00	10042	0,96
15	700861	0,00	0,74	0,48	0,15	0,58	0,78	17615	0,38
16	412528	0,13	0,22	0,58	0,08	0,15	0,17	10584	0,76
17	290378	0,57	0,09	0,20	0,07	0,15	0,00	7353	0,23
18	208059	0,83	0,20	0,18	0,19	0,16	0,00	5923	0,05
19	533511	0,49	0,45	0,63	0,50	0,72	0,05	12814	0,49
20	854236,5	0,00	0,16	0,54	0,01	0,08	0,00	17438	0,45
21	307323,5	0,51	0,91	0,83	0,84	0,80	0,04	7744	0,05
22	188190	0,82	0,18	0,32	0,21	0,37	0,23	5828	0,84
23	721216	0,56	0,91	0,94	0,81	0,88	0,11	16673	0,20
24	241159	0,00	0,94	0,10	0,15	0,94	0,93	7502	0,37
25	194575,5	0,19	0,22	0,16	0,40	0,32	0,02	6372	0,04
26	218460	0,51	0,27	0,44	0,30	0,44	0,36	6498	0,96
27	474012	0,12	0,01	0,00	0,01	0,01	0,00	10452	0,11
28	855600	0,00	0,12	0,03	0,47	0,20	0,32	23814	0,69
29	962200	0,10	0,13	0,43	0,09	0,29	0,40	20874	0,47
30	327125,5	0,00	0,08	0,50	0,03	0,21	0,35	8190	0,83
31	423054	0,45	0,51	0,55	0,42	0,39	0,13	12095	0,10
32	390792	0,26	0,63	0,95	0,59	0,91	0,20	9510	0,63
33	690308,5	0,00	0,00	0,00	0,00	0,00	0,00	20447	0,02
34	349308	0,58	0,83	0,78	0,88	0,84	0,22	8483	0,95
35	1121148	0,01	0,32	0,01	0,74	0,34	0,70	24973	0,65
36	178752	0,11	0,00	0,00	0,01	0,00	0,00	6268	0,27
37	214700	0,96	0,65	0,54	0,77	0,71	0,91	6625	0,29
38	401247	0,01	0,01	0,00	0,67	0,08	0,00	12925	0,46
39	307372,5	0,01	0,05	0,51	0,00	0,05	0,00	8063	0,77
40	308176	0,00	0,00	0,00	0,03	0,01	0,00	7865	0,89
41	486297	0,25	0,79	0,78	0,64	0,94	0,95	12040	0,06
42	275445,5	0,27	0,01	0,01	0,06	0,04	0,00	7353	0,19
43	478840,5	0,23	0,69	1,00	0,38	0,55	0,24	10677	0,71
44	571857	0,44	0,81	0,74	0,94	0,78	0,24	16571	0,73
45	393669	0,30	0,19	0,19	0,46	0,39	0,01	9440	0,15
46	379561	0,89	0,04	0,04	0,17	0,18	0,00	9434	0,96
47	464787	0,81	0,34	0,28	0,33	0,58	0,29	13860	0,47
48	258620	0,09	0,12	0,03	0,57	0,26	0,22	7493	0,88

49	529914	0,16	0,58	0,37	0,58	0,39	0,06	16167	0,02
50	522006	0,01	0,81	0,34	0,42	0,83	0,24	16502	0,94
51	1487992,5	0,00	0,97	0,12	0,21	0,78	0,85	33291	0,17
52	439845	0,32	0,69	0,76	0,44	0,56	0,67	10733	0,99
53	466306,5	0,00	0,00	0,21	0,00	0,01	0,00	12656	0,13
54	463133,5	0,39	0,01	0,00	0,12	0,03	0,00	11880	0,02
55	1006397	0,09	0,08	0,38	0,06	0,12	0,00	21780	0,19
56	265220	0,03	0,74	0,29	0,73	0,65	0,50	7430	0,70
57	474345	0,11	0,15	0,24	0,14	0,22	0,00	13764	0,07
58	341510	0,02	0,03	0,32	0,00	0,02	0,00	9316	0,12
59	980172	0,37	0,35	0,55	0,13	0,19	0,03	23883	0,01
60	332167,5	0,00	0,20	0,72	0,05	0,23	0,00	8246	0,00
61	622557,5	0,02	0,77	0,60	0,19	0,55	0,98	14229	0,88
62	289535	0,31	0,01	0,04	0,01	0,02	0,00	8107	0,50
63	235698	0,21	0,26	0,76	0,25	0,61	0,01	7119	0,34
64	316386	0,00	0,03	0,00	0,53	0,10	0,00	8319	0,64
65	514444	0,00	0,00	0,00	0,17	0,00	0,00	17370	0,01
66	293740	0,00	0,54	0,49	0,11	0,77	0,23	5614	0,93
67	357100,5	0,08	0,03	0,01	0,42	0,18	0,00	9387	0,40
68	293986	0,00	0,61	0,19	0,94	0,46	0,88	8456	0,92
69	208545,5	0,35	0,07	0,36	0,04	0,09	0,00	6272	0,55
70	262810	0,35	0,11	0,07	0,23	0,11	0,00	8894	0,80
71	284031	0,00	0,01	0,17	0,00	0,05	0,00	8607	0,01
72	288579,5	0,11	0,77	0,30	0,69	0,70	0,42	6888	0,80
73	417994,5	0,00	0,02	0,00	0,63	0,08	0,00	11466	0,29
74	232200	0,05	0,69	0,23	0,76	0,62	0,44	5885	0,47
75	412993,5	0,03	0,22	0,93	0,06	0,30	0,56	11610	0,93
76	206612	0,06	0,76	0,28	0,44	0,97	0,83	6498	0,92
77	264984	0,80	0,01	0,02	0,03	0,04	0,00	6383	0,75
78	276760	0,35	0,27	0,27	0,39	0,38	0,05	7360	0,97
79	816060	0,80	0,01	0,00	0,11	0,01	0,00	15225	0,00
80	257712	0,97	0,03	0,05	0,03	0,04	0,00	7547	0,90
81	293670	0,00	0,14	0,01	0,77	0,21	0,01	7680	0,14
82	647797,5	0,31	0,56	0,86	0,45	0,75	0,97	18285	0,91
83	274560	0,00	0,00	0,00	0,06	0,00	0,00	7946	0,00
84	752815	0,03	0,27	0,62	0,04	0,14	0,07	19293	0,67
85	397670	0,60	0,64	0,67	0,22	0,55	0,51	9238	0,99
86	952544	0,90	0,78	0,62	0,49	0,66	0,00	26928	0,08
87	929100	0,02	0,73	0,04	0,19	0,94	0,47	23814	0,58
88	479655	0,20	0,14	0,02	0,45	0,25	0,26	11448	0,79
89	484428	0,70	0,46	0,49	0,83	0,76	0,25	13750	0,66
90	375061	0,00	0,00	0,00	0,00	0,00	0,00	11303	0,00
91	1295775	0,97	0,04	0,16	0,04	0,08	0,04	46116	0,71
92	448362	0,58	0,17	0,27	0,06	0,19	0,00	10961	1,00
93	454537	0,02	0,16	0,01	0,91	0,25	0,00	9150	0,36
94	308100	0,22	0,27	0,06	0,55	0,18	0,00	9044	0,92
95	220765	0,02	0,16	0,38	0,03	0,12	0,00	5668	0,47
96	277455	0,00	0,01	0,00	0,10	0,01	0,00	8531	0,05
97	460728	0,23	0,71	0,35	0,93	0,73	0,00	9782	0,09
98	252006,5	0,72	0,00	0,00	0,00	0,00	0,00	7366	0,00

99	156584	0,54	0,16	0,19	0,40	0,42	0,00	5512	0,19
100	411340	0,58	0,44	0,66	0,26	0,62	0,54	8910	0,66

B.3. Difference value for Banerjee et al.

	Dwell time	Down-Down	Down-Up	Up-Down	Up-Up	Typing speed	Pause rate
1	0,22	0,40	0,35	0,35	0,35	0,20	0,95
2	0,18	0,31	0,33	0,28	0,33	0,30	0,68
3	0,22	0,38	0,38	0,42	0,41	0,16	0,89
4	0,15	0,29	0,34	0,30	0,34	0,19	0,89
5	0,23	0,49	0,43	0,40	0,49	0,28	1,05
6	0,26	0,66	0,75	0,64	0,74	0,15	1,41
7	0,15	0,47	0,49	0,48	0,47	0,19	1,38
8	0,16	0,53	0,53	0,49	0,52	0,11	1,38
9	0,16	0,39	0,52	0,47	0,48	0,11	1,26
10	0,25	0,56	0,47	0,54	0,55	0,23	0,89
11	0,24	0,40	0,39	0,37	0,37	0,38	0,87
12	0,19	0,40	0,44	0,49	0,45	0,25	0,97
13	0,30	0,47	0,47	0,35	0,38	0,40	1,07
14	0,18	0,35	0,42	0,36	0,41	0,19	0,93
15	0,25	0,39	0,40	0,33	0,38	0,19	0,95
16	0,31	0,38	0,37	0,35	0,39	0,22	0,97
17	0,26	0,37	0,49	0,48	0,53	0,34	1,17
18	0,28	0,43	0,48	0,44	0,46	0,30	1,06
19	0,34	0,41	0,44	0,43	0,43	0,13	1,01
20	0,23	0,38	0,36	0,36	0,39	0,31	1,20
21	0,18	0,44	0,49	0,47	0,50	0,21	1,34
22	0,23	0,43	0,45	0,39	0,37	0,32	1,05
23	0,24	0,31	0,29	0,31	0,31	0,29	0,82
24	0,42	0,48	0,54	0,51	0,52	0,18	1,39
25	0,18	0,38	0,43	0,39	0,41	0,20	0,75
26	0,22	0,38	0,47	0,31	0,47	0,19	1,00
27	0,19	0,41	0,42	0,40	0,44	0,25	1,04
28	0,15	0,39	0,38	0,38	0,41	0,23	1,08
29	0,22	0,47	0,48	0,49	0,46	0,14	1,10
30	0,32	0,52	0,40	0,53	0,56	0,21	1,44
31	0,18	0,38	0,44	0,37	0,40	0,16	0,98
32	0,23	0,36	0,41	0,34	0,35	0,22	0,80
33	0,27	0,39	0,45	0,38	0,35	0,60	0,86
34	0,35	0,42	0,42	0,48	0,47	0,11	1,13
35	0,13	0,28	0,32	0,36	0,32	0,14	1,01
36	0,10	0,54	0,57	0,64	0,65	0,37	1,30
37	0,20	0,41	0,37	0,34	0,44	0,13	0,85
38	0,22	0,41	0,43	0,46	0,45	0,35	1,11
39	0,20	0,40	0,41	0,43	0,43	0,32	1,07
40	0,22	0,51	0,57	0,56	0,59	0,20	1,23
41	0,24	0,42	0,47	0,37	0,36	0,16	0,97
42	0,24	0,43	0,51	0,45	0,40	0,24	1,11

43	0,17	0,33	0,40	0,33	0,32	0,22	1,02
44	0,15	0,46	0,44	0,35	0,38	0,13	0,89
45	0,16	0,36	0,45	0,37	0,39	0,27	0,85
46	0,19	0,53	0,55	0,42	0,48	0,09	1,33
47	0,27	0,47	0,40	0,44	0,46	0,36	1,11
48	0,13	0,47	0,41	0,39	0,51	0,13	1,24
49	0,19	0,41	0,42	0,41	0,41	0,26	1,24
50	0,13	0,31	0,41	0,36	0,40	0,30	0,91
51	0,31	0,36	0,37	0,38	0,39	0,14	1,08
52	0,18	0,44	0,45	0,48	0,40	0,25	1,16
53	0,21	0,32	0,40	0,30	0,32	0,24	0,87
54	0,14	0,29	0,44	0,27	0,40	0,13	1,10
55	0,14	0,26	0,31	0,28	0,30	0,30	0,91
56	0,30	0,49	0,54	0,50	0,50	0,07	1,11
57	0,17	0,41	0,52	0,40	0,53	0,36	1,07
58	0,20	0,42	0,36	0,38	0,33	0,43	0,87
59	0,24	0,45	0,40	0,39	0,35	0,23	1,10
60	0,22	0,38	0,39	0,45	0,37	0,29	1,23
61	0,30	0,38	0,37	0,39	0,42	0,30	1,14
62	0,25	0,46	0,39	0,46	0,46	0,52	0,99
63	0,28	0,40	0,32	0,49	0,37	0,41	0,96
64	0,35	0,43	0,42	0,36	0,40	0,18	1,05
65	0,25	0,38	0,49	0,38	0,40	0,24	0,95
66	0,37	0,49	0,57	0,48	0,55	0,17	1,19
67	0,16	0,26	0,33	0,31	0,38	0,26	0,88
68	0,27	0,46	0,54	0,48	0,51	0,22	1,11
69	0,27	0,34	0,35	0,26	0,41	0,22	0,91
70	0,33	0,46	0,44	0,38	0,42	0,32	0,98
71	0,28	0,47	0,45	0,46	0,50	0,21	1,04
72	0,23	0,51	0,43	0,49	0,51	0,12	1,31
73	0,26	0,42	0,48	0,35	0,43	0,14	1,00
74	0,24	0,37	0,45	0,42	0,44	0,32	1,15
75	0,13	0,35	0,36	0,36	0,42	0,32	0,94
76	0,26	0,40	0,50	0,45	0,48	0,31	1,02
77	0,23	0,39	0,37	0,35	0,37	0,23	0,77
78	0,15	0,50	0,52	0,42	0,44	0,11	1,16
79	0,29	0,35	0,34	0,40	0,42	0,24	0,88
80	0,39	0,52	0,47	0,42	0,55	0,36	1,14
81	0,26	0,39	0,44	0,42	0,45	0,21	1,14
82	0,24	0,34	0,45	0,38	0,45	0,19	1,14
83	0,47	0,55	0,69	0,56	0,70	0,31	1,39
84	0,15	0,31	0,36	0,35	0,39	0,17	0,78
85	0,17	0,24	0,28	0,25	0,24	0,08	0,83
86	0,19	0,24	0,27	0,24	0,26	0,23	0,84
87	0,24	0,36	0,39	0,34	0,33	0,11	0,83
88	0,29	0,46	0,58	0,47	0,51	0,27	1,14
89	0,19	0,49	0,44	0,46	0,51	0,20	1,24
90	0,76	0,94	0,73	0,95	0,92	0,40	1,01

91	0,18	0,40	0,37	0,30	0,39	0,19	0,76
92	0,23	0,50	0,47	0,46	0,49	0,13	1,25
93	0,24	0,38	0,42	0,45	0,41	0,22	0,95
94	0,18	0,38	0,39	0,40	0,45	0,28	1,24
95	0,27	0,40	0,41	0,38	0,46	0,12	1,27
96	0,27	0,41	0,52	0,53	0,51	0,43	1,03
97	0,29	0,39	0,41	0,46	0,49	0,37	1,06
98	0,09	0,53	0,56	0,51	0,60	0,32	2,00
99	0,20	0,40	0,42	0,32	0,45	0,33	1,00
100	0,22	0,37	0,29	0,32	0,36	0,30	0,90

C. Header for the .arff files for WEKA

```
@relation      user

@attribute     feature1       numeric
@attribute     feature2       numeric
@attribute     feature3       numeric
@attribute     class          {0,1}

@data
```