Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

Reidar Johannessen Strømme

Early gender detection using keystroke dynamics and stylometry

Master's thesis in Master in Information Security
Supervisor: Patrick Bours
June 2021

NTNU
Norwegian University of Science and Technology

Reidar Johannessen Strømme

Early gender detection using keystroke dynamics and stylometry

Master's thesis in Master in Information Security
Supervisor: Patrick Bours
June 2021

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

NTNU
Norwegian University of
Science and Technology

Abstract

For many people, the Internet has become an important arena for meeting new people. In chat conversation with strangers, one does however not have any guarantees that the conversation partner is the one he/she claims to be. Gender is one example of something a conversation partner can provide false information about. Earlier research has achieved good results regarding gender detection based on complete conversations. In this project we explored the possibilities of detecting the gender earlier in the conversation by using machine learning models trained with keystroke dynamics and stylometry features.

We achieved promising results and found clear indications that early gender detection should be possible, without much accuracy loss. Based on complete chat conversations, where the average length is 28 written messages from each participant, we were able to achieve an accuracy of 80%. We experienced no accuracy loss when basing the classification on half conversations (14 messages). When basing the classification on only 5 messages (approximately 18% of the length of complete conversations), the accuracy loss was still <5 percentage points.

Sammendrag

For mange mennesker har Internett blitt en viktig arena for å stifte nye bekjentskap. Dette innebærer ofte at man befinner seg i chatsamtaler der man ikke har noen garantier på at motparten er den som vedkommende utgir seg for å være. Kjønn er et eksempel på noe en samtalepartner kan oppgi falsk informasjon om. Tidligere forskning har oppnådd gode resultater på å oppdage det sanne kjønnet til en chatsamtalepartner basert på hele samtaler. I dette prosjektet forsøker vi å finne ut om det er mulig å oppnå dette på et tidligere tidspunkt i samtalen ved å trene opp ulike maskinlæringsmodeller med data basert på tastefrekvens og stilometri.

Vi oppnådde lovende resultater og fant klare tegn på det skal være mulig å oppdage kjønnet til en chatsamtalepartner tidlig i samtalen, uten store tap av treffsikkerhet. Basert på hele samtaler, hvor gjennomsnittlig lengde var 28 sendte meldinger per deltaker, oppnådde vi en treffsikkerhet på opptil 80%. Ved halverte samtalelengder (14 meldinger) oppsto det ingen tap av treffsikkerhet. Ved å redusere antall meldinger til 5 (omtrent 18% av hele samtalelengder) var tapet av treffsikkerhet fortsatt <5 prosentpoeng.

Contents

Figures

Tables

Acronyms

AiBA Author input Behavior Analysis. v, vii, 16, 17

CNN Convolutional Neural Network. 9, 15

DT Decision Trees. 10, 15

k-NN k-Nearest Neighbors. viii, ix, 6, 9, 14, 15, 31, 33, 34, 36, 41–43, 45, 53, 56, 69–72, 75

KD Keystroke dynamics. 15, 34, 49, 56

LatPP Press-press latency. 5, 6, 14, 19, 20, 22, 23, 25, 26

LatPR Press-release latency. 5, 6, 14, 19, 20, 22, 23, 25, 26

LatRP Release-press latency. 5, 6, 14, 19, 20, 22, 23, 25, 26

LatRR Release-release latency. 5, 6, 14, 19, 20, 22, 23, 25, 26

LB LogitBoost. 7, 10, 14, 15

LogR Logistic Regression. 6, 7, 9, 10, 14, 15

MLL Multi-nomial Log Linear. 7, 14

MLP Multi-Layer Perceptron. 6, 14

MRMR Minimum Redundancy Maximum Relevance. 27

NaN Not a number. 20, 24

NB Naïve Bayes. 6, 7, 9, 10, 14, 15

NN Neural Network. 31

NTNU Norwegian Unisversity of Science and Technology. 16, 17

RBFN Radial Basis Network Function. 6, 14

RF Random Forest. vii, viii, 6, 7, 9, 10, 14, 15, 31, 33, 34, 36–40, 42, 43, 45–53, 56–58, 66–69, 75

SVM Support Vector Machines. ix, 6, 7, 9, 10, 14, 15, 31, 33, 34, 36, 42, 43, 45, 53, 56, 72–75

Chapter 1

Introduction

1.1 Topics covered by the project

The goal of this project is to explore the possibilities of early gender detection in chat conversations. To achieve this, we will mainly focus on two topics. The first one of these is keystroke dynamics, which is the act of recognizing people based on the way they type on a keyboard, most often looking at how long each key is pressed and how much time passes between each keystroke. The second is stylometry, which involves determining the author of a text based on the style of writing, often consisting of aspects such as punctuation usage, frequency of certain words/phrases or similar. Both of these topics involves determining who is the author of a given text, and used in combination, it can reveal much information about the author. This project will focus on how analyzing a text, specifically chat logs, using keystroke dynamics and stylometry can reveal information about the author's gender, and more specifically, to analyze how many chat messages is needed before an accurate decision can be made.

1.2 Keywords

Keystroke dynamics, stylometry, soft biometrics, behavioural biometrics, biometric fusion, gender detection, gender classification.

1.3 Problem description

When talking to strangers online, you cannot be completely certain that the person you are talking to is who he/she claims to be. This project aims to remove some of this uncertainty by trying to determine the gender of the person you are talking to, based on keystroke dynamics and stylometry. Research until now has mainly focused on determining the gender of a person based on all messages one have written in a chat conversation, meaning that the classification is performed after the conversation has ended. When talking to a stranger, most people would

however prefer to find out if the conversation partner lies about his/her identity as soon as possible and before the conversation has ended. This project aims to address this problem by finding out to what extent it is possible to classify the gender of a person earlier in the conversation.

1.4 Justification, motivation and benefits

By knowing the true gender of the person you are talking with, the Internet in general could become a safer place. There are many harmful situations that could have resulted in different outcomes if this was the case. One example is "Sandrasaken", which at the time was the biggest child exploitation case in Norwegian history [1]. An adult male claimed to be a younger female and tricked and blackmailed hundreds of young boys to send indecent images and videos. In another case [2], a young man was being blackmailed by someone claiming to be a young woman who possessed indecent videos of him. It ended with the young man committing suicide. Another famous example is the "Meier-case", where a 13-year-old girl was cyberbullied by a group of teenage girls, and one adult woman, posing as a nice young boy who eventually started to send cruel and harassing messages after first gaining her trust [3]. This case also ended with a suicide of the 13-year-old girl. A final example is the case of love scams. These are cases where criminals use a false identity and pretend to initiate romantic relationships with unsuspecting victims. After trust has been established, they start asking for increasing amounts of money. It is not uncommon for victims to lose millions of Norwegian crowns (NOK) due to this [4].

All of these cases may have been avoided if the victims were aware of the deceptive nature of the person they were talking with. This makes early gender detection a beneficial tool for both reducing certain forms of cybercrime and making online platforms safer.

1.5 Research questions

The sections above resulted in the main research question ***Is it possible to accurately classify the gender of a person early in a conversation using keystroke dynamics and stylometry?*** In relation to this, the following associated sub-questions have also been defined:

How much accuracy is lost when performing the classification early in a conversation? - One could expect that the accuracy will be lower when basing the classification on a lower number of messages than on complete chat conversation logs. To determine to which extent early gender detection is possible, we need to find out how big this accuracy loss is. If the accuracy loss grows too high, the usefulness of early gender detection would be vastly reduced as one cannot trust the classification.

How early is it possible to perform the classification while maintaining accuracy? - The usefulness of early gender detection would increase the earlier the classification is made. This does however only hold if there are no significant degradations in accuracy. Finding out how early it is possible to perform the classification while maintaining accuracy would allow us to maximize the usefulness of early gender detection.

When in a conversation should the classification be made? - The performance of early gender detection is based on the two criteria accuracy and the number of messages needed before the classification is performed. These are expected to conflict to some degree as obtaining a high accuracy will often depend on a large number of messages, while using a small number of messages will often result in lower accuracy. Finding the optimal moment to perform the classification would hopefully allow us to preserve both these performance measures, which is crucial for the usefulness of early gender detection.

How should stylometry and keystroke data be fused? - When using features from two modalities, a process known as biometric fusion (see Section 2.3) is necessary. The method of fusion can affect the overall accuracy of the classification and finding the best fusion method is thus necessary to find a proper answer to the main research question.

Chapter 2

Related work

This chapter will cover the current state-of-the-art regarding how stylometry and keystroke dynamics have been used for gender detection up until now. Within the topic of predicting gender using stylometry and keystroke dynamics, the majority of existing research only considers either stylometry or keystroke dynamics, while there is generally less literature about how to use these modalities in combination. This makes it natural to divide this chapter into three main sections, where each section covers one of the above cases. In addition, there will be one final section where we will discuss existing research regarding how one could at which point in the conversation the classification should be performed.

2.1 Keystroke dynamics

2.1.1 Introduction to keystroke dynamics

Biometrics has often been used for authentication [5]. Contrary to traditional methods of authentication, such as passwords, PINs or key cards, biometric authentication is based on "what you are". This includes physical characteristics, such as fingerprint recognition [6] or face recognition [7], or characteristics linked to ones behaviour, such as voice recognition [8] or signature recognition [9]. Keystroke dynamics is another one of these behavioural characteristics, and refers to the way one types on a keyboard [10, 11]. This can include features such as how long keys are held down, how long the pauses between keystrokes are or pressure when pressing down a key [10]. These features vary from person to person in such degree that it can be used to distinguish people from one another, merely by the way they type [10, 11]. Because of this, keystroke dynamics has often been used for authentication. One example could be that if your computer is stolen, and the thief also knows your password and tries to log in, the computer could deny the thief access because his keystroke rhythm when typing the password (most likely) deviated from yours. Because keystroke dynamics is able to distinguish between persons, one could wonder if keystroke dynamics also could distinguish between groups of people that share a certain trait, for example right-

handed and left-handed people, males and females, children and adults or similar. These non-unique characteristics are called soft biometrics [12]. This section will cover research that has been done on distinguishing keystroke dynamics between genders (male or female).

2.1.2 Keystroke dynamics gender classification

To correctly classify the gender of a person, a necessary prerequisite is that there exists keystroke dynamics features that are able to distinguish male and female typists. Various approaches to determine this have been taken in literature, with varying conclusions being drawn. One example is that [13] concluded that females generally types faster than males, while [14] concluded that there is no such difference. As this section will show, the consensus does however seem to be that there exists at least some difference between males and females, in regard to keystroke dynamics.

In keystroke dynamics, the potential features are generally extracted from a dataset containing key values and accompanying timestamps for when they were pressed and released [10, 11]. The features will then consist of timing relations between different keys. The most common is to have timing relations between 2 keys, in other words between bigrams [10, 11]. The possible features are then:

- Press-Press latency (LatPP) - Time between press of first key and press of second key.
- Press-Release latency (LatPR) - Time between press of first key and release of second key.
- Release-Press latency (LatRP) - Time between release of first key and press of second key.
- Release-Release latency (LatRR) - Time between release of first key and release of second key.
- Duration - Time between press and release of the same key.

It is also possible to extract features from segments longer than bigrams (often called n-grams, where n is the length of the sequence), but this is less frequently encountered. There have been several different feature sets used in research, but most variants include a combination of the features listed above, sometimes with small deviations.

In [13], the features were LatPP, typing speed, number of keystrokes in a message, total duration of written text (time between first and last keystroke) and total duration of time spent using the backspace key. In addition, some stylometry features were used, which will be covered in Section 2.2. The feature set in [14] considered all bigrams appearing more than 3 times and extracted the features LatPP and both durations. In [15], the feature set consisted of durations for each key, LatRP, n-gram latency (mean time of n consecutive key presses, where $2 \leq n \leq 4$), standard deviations for the preceding features, relative frequency of deletions, total number of keystrokes divided by the number of characters in the

final text and a final feature which they defined as LatRP + duration of second key in bigram.

The feature set in [16] consisted of 6 features for each of the 20 most frequently used bigrams. The features were durations of the two keys, LatRP, LatPP, LatRR and LatPR. In addition, they used deletion ratio (number of deleted characters divided by total number of typed keys) and average thinking time (time between two sent messages) as two additional features.

The feature set used in [17] and [18] consisted of, for every bigram, LatPP, LatPR, LatRP and LatRR. In [19], they used LatPP and LatRP, but also the pressure and finger area (area of screen the press occurred) of keystrokes. This was possible due to the data collection being performed on touchscreen keyboards on smartphones. The features in [20] consisted of durations for each keystroke and durations for specific groups of keys (numbers, letters, special characters etc.).

Finally, in [21], the features were LatRP and durations of each individual key and certain groups of keys. The key groupings were based on which finger/hand they are typed with, which row on the keyboard the keys are in and the whether the key value was common/rare. In addition, several stylometry features and features relying on both stylometry and keystroke dynamics were used. These are discussed in Section 2.2.

After the relevant features have been extracted from a variety of subjects (both male and female), the next step is to use the data to create a model that can accurately classify the gender of a typist. For the last several years, this classification is most often done utilizing various machine learning concepts.

The highest accuracy was found in [13], which on a dataset consisting of chat logs from 25 males and 35 females, achieved an accuracy of 98.3% with a Random Forest (RF) classifier using leave-one-out cross-validation. Details about Random Forest classification can be found in [22].

In [14], a collection of several classification models was used on a dataset consisting of keystrokes from 39 females and 36 males. The keystrokes were logged from everyday use. The models were Support Vector Machine (SVM), RF, Naïve Bayes (NB), Multi-Layer Perceptron (MLP) and Radial Basis Function Network (RBFN). The reasoning behind this diverse collection of classification models was to form an opinion on how their performances compared to each other. They found that all of them performed well, but RBFN was the best with an accuracy of 95.6% using 10-fold cross-validation. Details about Support Vector Machines, Naïve Bayes, Multi-Layer Perceptron and Radial Basis Function Network can be found in [23–26] respectively.

A similar approach was used in [15], but with a different set of classification models. On a dataset consisting of freely written texts from 1519 subjects (997 females and 522 males), the models Logistic Regression (LogR), SVM, k-Nearest Neigbours (k-NN), C4.5 and RF were used. SVM, k-NN and RF performed best for gender recognition, all with an accuracy of 73% using 10-fold cross-validation. Details about Logistic Regression, k-Nearest Neighbors and C4.5 can be found in [27–29] respectively.

In [16], they aimed to classify the gender based on chat logs between 10 females and 35 males. Some male participants were then removed to make a balanced training set. Separate RF classifiers were used on each of the 20 selected bigrams, and then the generated scores were fused to classify the gender of the author of a single chat message. They performed majority voting on all messages by the conversation participant to perform the final gender classification. This achieved an accuracy of 76% using 3-fold cross-validation.

Another approach, as described in [20], used only an SVM classifier. On a dataset containing 121 users (53 females and 68 males), they achieved an accuracy of 63.29% using 5-fold cross-validation. A similar approach was used in [17], but with higher accuracy (84% at most). In [18], there was again used a collection of different classification models. The dataset consisted of typings of a short static text by 21 females and 71 males. By using the models SVM, NB, RF and Multi-nominal Log-Linear (MLL), they found RF to be the most accurate with an accuracy of 62.63% using a 50/50, training/testing ratio. Details about Multi-nominal Log-Linear can be found in [30].

In [21], the analysis was performed on a dataset consisting of texts written freely, in response to some given questions, by 567 males and 415 females. For classification they then tried the classifiers LogitBoost (LB), NB, SVM and LogR. Best accuracy (51.6%) was achieved with LB using 10-fold cross-validation. Details about LogitBoost can be found in [31]. Finally, in [19], on a dataset consisting of keystrokes from 24 males and 18 females, it was used an RF classifier, which achieved an accuracy of 64.76% using leave-one-user-out cross-validation. The findings are summarized in Table 2.1.

2.2 Stylometry

2.2.1 Introduction to stylometry

Stylometry refers to the analysis of which style a text is written in [32, 33]. People tend to write in their own distinct style, which can be shaped from several factors such as mood, education level, age, gender, dialect or whether one is a native speaker or not [33]. All these factors in combination lead to people making certain linguistic choices [32, 33]. Examples could be that a university professor might use a complex and varied vocabulary, a child might make many common spelling mistakes, a teenager might use more slang and other hip phrases and a person who is excited/angry/frustrated might use more exclamation marks (!) and upper-case characters. This has led to two main use-cases, author attribution/verification and author profiling [32, 33]. Author attribution/verification means to verify whether a text was written by a particular author and author profiling means to analyse whether a text reveals information about the author such as age, gender or level of education [32, 33]. Like keystroke dynamics, stylometry can also be defined to be a behavioural biometric characteristic. This section will focus on how stylometry can be used for author profiling in regard to gender detection.

2.2.2 Stylometry gender classification

As with keystroke dynamics, the process of determining the gender (male or female) of a text's author, relies on the fact that there exists stylometry features that are able to distinguish males from females. In regard to stylometry, the potential features are whatever you can extract from a written text. In general, there are however three main categories of features that are used, which are phonetic features, lexical features and syntactic features [32]. Phonetic features involve features based on single characters or syllables. Examples could be count of certain characters, ratio of vowels/consonants or count of certain syllables. Lexical features involve features based on word choice, some examples being use of dialect words, average word length or number of unique words. Syntactic features involve features in regards to sentences. Examples could be tendency to use complete sentences, sentence length or use of certain linguistic concepts (e.g., chiasms or parallel syntax). In research, several variants have been used.

Some approaches use relatively simple features. Examples could be as in [16], where only two stylometry features are used, namely average length of words and average number of words in a message. Another example is [13], where the stylometry features consisted of merely the length of messages, the density of various characters and word count. This approach did however also use keystroke dynamics features, as discussed in Section 2.1. In [34], the features consisted of message length, average word length, character frequency, number of distinct words and usage of emojis, punctuation and stop words. Such simple feature sets have the advantage of being easy to extract and pre-process. An equally simple feature set was used in [35], which consisted of character counts (for special characters, spaces and punctuation), count of different kinds of emojis, average length of words and total text length. A shared characteristic among all of these is that features were extracted from a dataset based on chats or tweets, which in general consists of short texts. This limited amount of text could make it difficult to extract complex features.

One commonly held belief is that males and females differ in the way they express emotion. This theory was tested in [36], where the feature set consisted of word frequencies and metrics regarding the usage of emotion-based words such as "happy", "love", "sorrow" and "misery". This strategy deviates from many of the other approaches which focuses on more general features. The features in [37] consisted of many of the same features already mentioned, like the frequency of certain words and characters and count of different punctuation symbols. It did however also use vocabulary richness and frequency of multi-media content (possible due to the features being extracted from tweets, which allows posting of such content). Many of the same features were also used in [38], which used the features of vocabulary richness, count and ratio of punctuation symbols and length/count of words and sentences.

Other research have followed the philosophy that the more features the better, and thus ended up with rather complex feature sets. One example is in [39], where

a total of 545 features were used. These consisted of count of certain characters, count of certain words/phrases, vocabulary richness, frequency distribution of word lengths and features regarding message structure (paragraph length, use of greetings, correct punctuation etc.). Similar complexity is found in [40], where the features were word length, number of special characters and whether they were repeated (e.g., !!!! or ???), average number of words in a sentence, vocabulary richness, sentence richness (whether sentences tended to be complete) and usage of words/phrases from different categories (e.g. greetings, profanity and emotion based words). Finally, in [21] the stylometry features consisted of various metrics for sentences, words, character types and punctuation, in addition to vocabulary richness. This approach also used keystroke dynamics features as mentioned in Section 2.1 and some features that are derived from both keystroke dynamics. They call this language production features and consists of features such as latency between words of different categories (nouns, verbs, singular/plural etc.) or word count within a writing burst (a writing burst is a sequence of keystrokes with short pauses).

Following in the same manner as with keystroke dynamics, the next step is to train a model with text data from males and females which goal is to classify the gender of the author accurately. As with keystroke dynamics, this is most often done using machine learning concepts as this generally yields good results [33]. The differences between the approaches found in literature is then generally which machine learning model is used and the properties of the dataset it is used upon.

In [16], they aimed to classify gender based on chat logs between 10 females and 35 males, where some male participants were removed to make a balanced training set. An RF classifier was used, which gave an accuracy of 64% using 3-fold cross-validation. A dataset derived from chat logs was also used in [34]. The dataset consisted of chat logs from 200 male and 200 female profiles. Using an NB classifier, an accuracy of 84.2% was achieved using 10-fold cross-validation. A k-NN classifier was also tested, but with poorer results (accuracy of 64.6%). A final approach using chat data is found in [13]. In [13], on a dataset consisting of chat logs from 25 males and 35 females, they achieved an accuracy of 98.3% with an RF classifier using leave-one-out cross-validation.

Tweets from Twitter has also been shown to be a popular source of datasets. In [35], on a dataset consisting of tweets from 1030 males and 1030 females, a collection of different classifiers was used, consisting of LogR, RF, SVM and NB. On the testing data, they managed to achieve an accuracy of 76.52%, but they do not mention which of the classifiers this was achieved by. In [37], they used a dataset consisting of tweets from 486 males and 514 females for training. By using an SVM-classifier, they achieved an accuracy of 83.16%. In [40], they achieved an accuracy of 97.7% using a Convolutional Neural Network (CNN). They do however not share details about the dataset, other than it consisted of tweets. This makes it difficult to assess how impressive that accuracy is. Details about Convolutional Neural Networks can be found in [41].

Other sources of data have also been used. In [38], the training dataset consisted of text extracted from 328 male and 151 female Facebook-profiles. They tried the classifiers J48, RF, SVM and NB. They found RF to be the best with an accuracy of 81.3%, using 10-fold cross-validation. Accuracies achieved by the other classifiers were not disclosed. Details about J48 can be found in [42].

In [39], two different datasets were used. One of them consisted of 3474 news articles written by males and 3295 written by females. The other consisted of 4947 e-mails written by males and 4023 written by females. They achieved accuracies of 76.75% and 82.23% on the two datasets respectively, using SVM with 10-fold cross-validation. They also tried the classifiers LogR and Decision Trees (DT), but this resulted in lower accuracies. Details about Decision Trees can be found in [43].

In [21], the analysis was performed on a dataset consisting of texts written freely in response to some given questions by 567 males and 415 females. For classification they then tried the classifiers LB, NB, SVM and LogR. Best accuracy (51.6%) was achieved with LB using 10-fold cross-validation. Finally, in [36], on a dataset consisiting of journal entries from 43 males and 43 females, a maximum accuracy of 91.8% was achieved using SVM with 10-fold cross-validation. Table 2.2 summarizes the findings. One interesting observation in Table 2.2 is that more complex stylometry features does necessarily imply increased accuracy.

2.3 Combining stylometry and keystroke dynamics

2.3.1 Introduction to fusion

When using features from more than one biometric characteristic/modality, so-called multi-modal biometrics, the process of biometric fusion is a necessity. In general terms, biometric fusion involves taking input from different sources into account, with the goal of making more accurate decisions with higher confidence [44, 45]. As an example, the idea is that if a face recognition system claims that a subject is John Doe and a fingerprint recognition system claims that a subject is John Doe, one can be more confident that the subject actually is John Doe than if only one of the systems claimed so.

There are in general 5 approaches to biometric fusion, which are distinguished by where in the biometric process they take place [44, 45].

- Sensor-level fusion: Combine biometric data from multiple sensors before features are extracted.
- Feature-level fusion: Combine several feature sets (from the same subject) into one extended feature set.
- Score-level fusion: Process the feature sets individually and combine the resulting score into a final score.
- Rank-level fusion: Create a ranking of scores in descending order for each subsystem. The option with highest rankings is chosen.

- Decision-level fusion: Process each feature set individually. The decisions of all subsystems are combined to make a final decision.

It is hard to call one approach better than the others, but some claim that score-level fusion generally tends to perform best [44].

2.3.2 Fusing stylometry and keystroke dynamics

Based on the conducted literature review, there exist only limited research about fusing stylometry and keystroke dynamics. The only found research that uses both keystroke dynamics and stylometry for gender detection is found in [13], [16] and [21].

In [16], they used score-level fusion with equal weights assigned to the scores from the stylometry classifier and keystroke dynamics classifier. This did however not increase accuracy. The accuracy remained at 64%, which was the same as using the stylometry classifier by itself. The keystroke dynamics classifier achieved an accuracy of 72% by itself. In [13] and [21], they did not explicitly state how the fusion was performed. Based on the way the features were presented, it does however seem likely that a feature-level approach was used. In [13] and [21], they achieved accuracies of 98.3% and 51.6% respectively. More details about the specific features and classification can be read in Section 2.1 and Section 2.2.

Some approaches of fusion are however not relevant in regard to gender detection in chats using keystroke dynamics and stylometry. Sensor-level fusion cannot be used because the data collection is not performed by multiple sensors. Rank-level fusion is also not relevant as it is mainly used for identification. In addition, decision-level fusion can be challenging as there are only two modalities that are to be fused. This can result in a tie when the two modalities disagree on whether the chatter is male or female. Some sort of tiebreaker would thus be needed. This makes score-level and feature-level fusion the most relevant for use in this project.

In conclusion, the amount of research regarding the fusion of stylometry and keystroke dynamics, is rather limited. An important aspect of the project will therefore be to determine how the fusion should be performed.

2.4 Introducing gender levels

When performing a classification task, it is not always the case that the decision made by the classifier is correct. The assigned class is only determined to be the most probable one, based on the data the classifier has received. The topic of early gender detection implies that the classifier would need to base the decision on a relatively low number of messages, and not complete conversations. This can make it more challenging to perform correct classifications. In addition, the perceived gender of a person could change during the course of a conversation as the classifier receives more messages to base the decision upon. Even if a person's first message is classified as male, it does not necessarily imply that the person's

true gender is male. A decision made at a later point will thus probably be more trustworthy, in the sense that it is more likely to be correct.

As an example, consider a person's sent chat messages at times t_0 to t_3, where the gender classification of the sent messages at time t_i is displayed in Figure 2.1. For the sake of simplicity, assume that all message classifications are weighted equally.

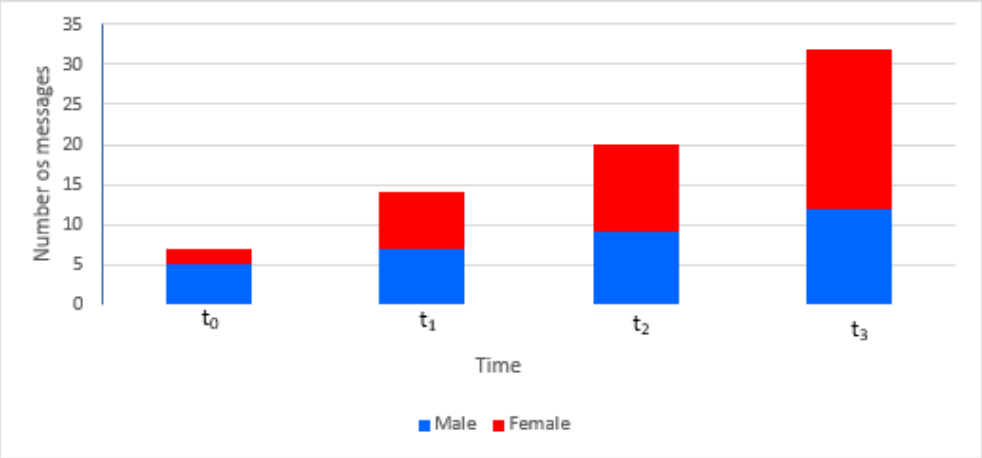

Figure 2.1: Sent chat messages from a chat conversation participant at different times with associated gender classifications

At time t_0, the person has sent 5 messages that has been classified as male and 2 messages has been classified as female. At time t_1, there has been sent an equal number (7) of male and female messages. At time t_2, there has been sent 9 messages that has been classified as male and 11 messages has been classified as female. At time t_3, the person has sent 12 messages that has been classified as male and 20 messages has been classified as female.

It should be clear that a final gender classification made at t_3 is more likely to be correct than the decisions made at t_0, t_1 and t_2. At t_3, more data is available, and it shows a clear trend that most messages are considered to be female. As a result, t_3 is the first point where it is possible to make a somewhat confident final classification. As we described in Section 1.5, one of the key aspects regarding early gender detection is to know when the classification can be made. It is thus necessary with a system that is able to determine when the final classification can be made with a sufficient confidence.

This has not yet been suspect to much research, but there has been research dealing with same issue within the area of continuous authentication.[1] Many of these results can potentially also be used for early gender detection.

Continuous authentication systems sometimes rely on trust levels to ensure that genuine users are not being rejected by the system. The user should only be

[1] A way of authentication where the authentication process is performed continuously. Done to ensure that the user is genuine even after the entry-point authentication has been completed.

rejected when the system is somewhat certain that user is not genuine. This can be solved by using a penalty-and-reward system [46, 47]. When a user enters the system, the trust is set to the maximum level. For each action the user performs, the trust level is re-evaluated. If the action is deviating from the user's normal pattern, the trust level decreases (penalty) and if the action is considered normal, the trust level increases (reward). If the trust level decreases to a certain threshold, the user can be rejected with high confidence that he/she is not genuine.

A similar system can be imagined for the purpose of early gender detection. One could consider an axis between 0 and 1, where 0 would represent complete certainty that a conversation participant is male and 1 would represent complete certainty that a conversation participant is female. By setting the default gender level to 0.5, one could increase or decrease the value based on whether the next message is classified as male or female. The value adjustments could be either fixed or varying, as discussed in [46, 47]. When the value is approaching further away from 0.5 towards defined thresholds, the gender classification of the conversation participant would most likely have a higher probability of being correct. Using the same message classifications used in Figure 2.1, one could imagine the gender level being adjusted as in Figure 2.2. This could solve the issue presented earlier in this section as a final decision would not be taken before it is possible to perform the classification with a certain confidence.

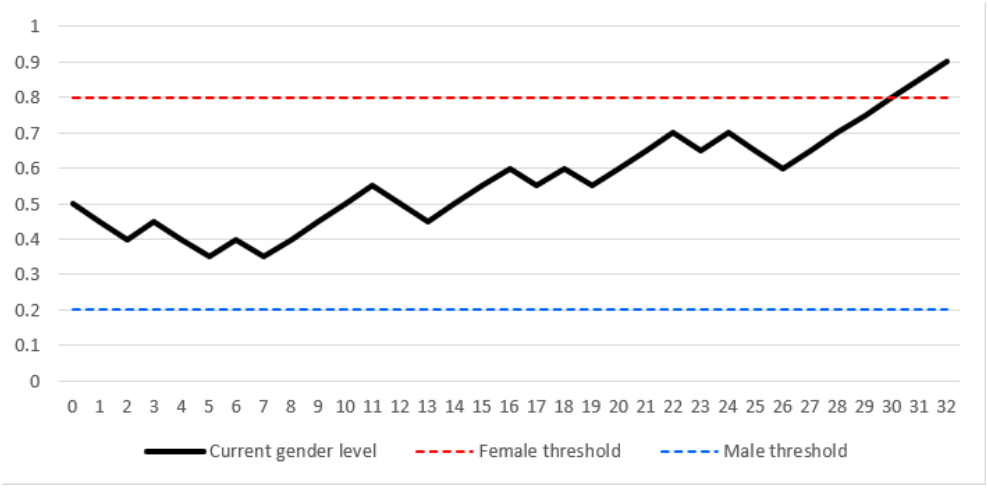

Figure 2.2: A visualization of how the gender level could adjust after each processed message

Reference	Features	Dataset (males+females and source)	Classifier	Accuracy
[13]	LatPP (median), typing speed, duration of messages, total duration of backspaces + stylometry	25+35, from chat logs	RF	98.3%
[14]	LatPP, durations	39+36, from everyday use	SVM RF NB MLP RBFN	85.1% 81.9% 78.6% 85.1% 95.6%
[15]	Durations, LatRP, n-graph latency, standard deviations, deletion ratio, number of keystrokes divided by number of characters	522+997, from free-text	LogR C4.5 SVM k-NN RF	69% 67% 73% 73% 73%
[16]	LatPR, LatPP, LatRP, LatRR, durations, deletion ratio, thinking time	35+10, from chat logs	RF	72%
[16]	Same as above + stylometry	35+10, from chat logs	RF	64%
[17]	LatPP, LatPR, LatRP, LatRR	78+32, from free-text	SVM	84%
[18]	LatPP, LatPR, LatRP, LatRR	71+21, from fixed text	SVM NB MLL RF	59.33% 52.48% 50.12% 62.63%
[19]	LatPP, LatRP, finger area, pressure	24+18, from performing fixed tasks on smartphone	RF	64.76%
[20]	Durations	68+53, from both free- and fixed text	SVM	63.29%
[21]	LatRP, durations + stylometry features and combination features	567+415, from free-text	LB SVM NB LogR	51.6% 46.2% 46.8% 51.3%

Table 2.1: Summary of gender detection with keystroke dynamics

Reference	Features	Dataset (males+females and source)	Classifier	Accuracy
[13]	Message length, density of various characters + KD	25+35, from chat logs	RF	98.3%
[16]	Length and number of words	35+10, from chat logs	RF	64%
[16]	Same as above + KD	35+10, from chat logs	RF	64%
[21]	Metrics for words, sentences, character-types and punctuation, vocabulary richness + KD and combination features	567+415, from free-text	LB SVM NB LogR	51.6% 46.2% 46.8% 51.3%
[34]	Length of words and messages, character frequency, number of distinct words, usage of emojis, punctuation and stop words	200+200, from chat logs	k-NN NB	64.6% 84.2%
[35]	Character count, emoji count, word and text length	1030+1030, from Twitter	LogR SVM NB RF	76.5% 76.5% 76.5% 76.5%
[36]	Word frequency, usage of emotion-based words	43+43, from journal entries	SVM	91.8%
[37]	Word and character frequency, count of punctuation symbols, vocabulary richness, frequency of multi-media content	486+514, from Twitter	SVM	83.16%
[38]	Vocabulary richness, sentence length/count, word length/count, count/ratio of various characters	328+151, from Facebook profiles	RF J48 SVM NB	81.3% - - -
[39]	Character counts, count of certain words/phrases, vocabulary richness, frequency distribution of word length, message structure	3474+3295, from news articles	SVM LogR DT	76% 67% 70%
[39]	Same as above	4947+4023, from e-mails	SVM LogR DT	82% 71% 72%
[40]	Word length, usage of special characters and certain words/phrases, word count in each sentence, vocabulary and sentence richness	Not revealed	CNN	97.7%

Table 2.2: Summary of gender detection with stylometry

Chapter 3

Data collection

The methodology of this project consists of two main parts, data collection and data analysis. This chapter will focus on the former by describing how the data collection was performed and highlighting the structure and properties of the obtained dataset. This dataset will be subject to the data analysis described in Chapter 4.

3.1 AiBA

The dataset used in this project was obtained through the AiBA project. AiBA[1] is a current research project conducted by NTNU, with the goal of developing tools and solutions to help protect children from sexual predators. They do this by detecting "cyber grooming", which is the process where adults contacts children with the end-goal of arranging inappropriate physical meetings or luring them to send inappropriate images/videos. In situations like this, it is not uncommon that the perpetrator uses a false identity by lying about their age and/or gender. AiBA aims to combat this by creating systems that are able to detect the true age and gender of a person automatically.

To be able to do this, they have collected a dataset consisting of chat data that can be used for training such systems. They created a chat-service where anyone (above the age of 18) can register and be paired up with a stranger anonymously to chat with using their own devices. A screenshot of the chat interface is seen in Figure 3.1. During the course of the conversations, the messages and keystroke actions were recorded and were labelled with gender and age.

During the registration, the participants also selected which language they would prefer to chat in. The viable options were either Norwegian, English or both. If both languages were selected, two separate accounts were generated where each of them would be used for one language. In the analysis part of this project, we will primarily use the Norwegian part of the dataset. We will mainly focus on one language because features will not necessarily translate very well from one

[1] https://www.aiba.ai

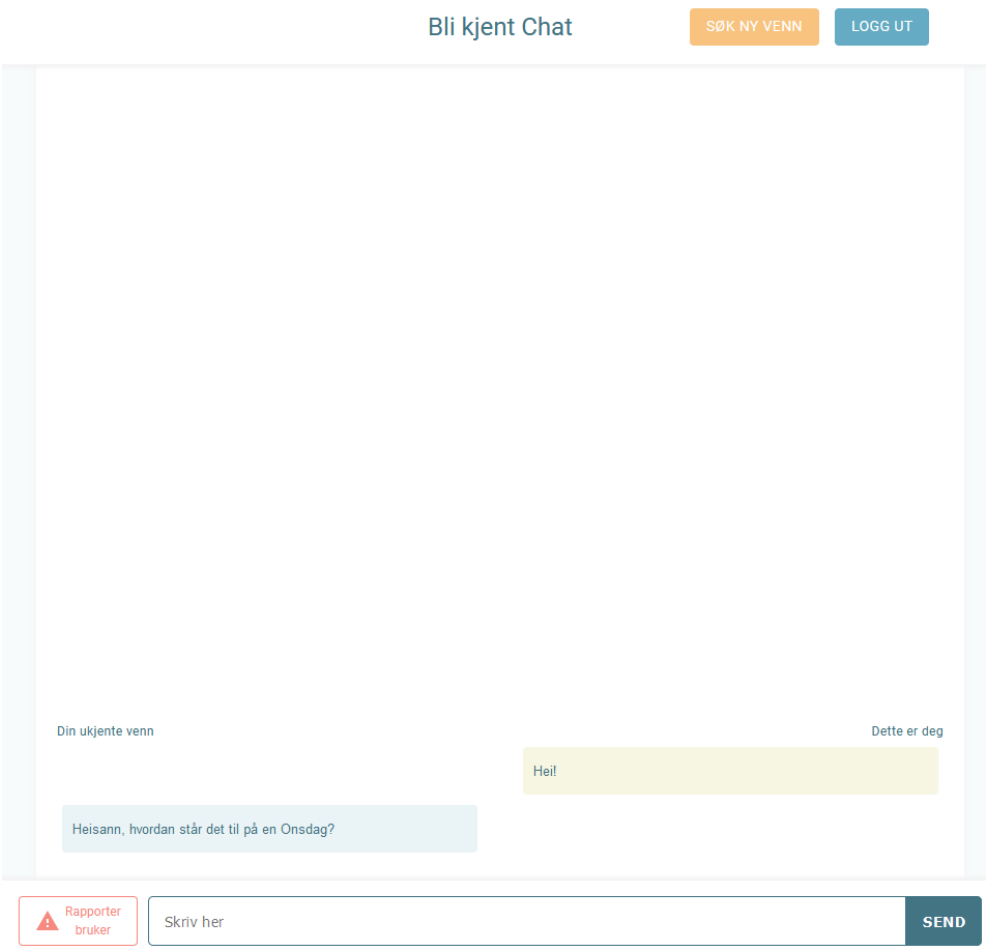

Figure 3.1: Screenshot of the AiBA chat interface

language to another. The Norwegian part of the dataset was selected due to the following reasons:

1. The Norwegian part of the dataset is much larger than the English one (containing more messages and keystrokes);
2. Norwegian is the native language of the author of this thesis, which makes the Norwegian part of the dataset easier to work with; and
3. This project is performed in association with NTNU, which is after all a Norwegian university.

This dataset makes it possible to extract both keystroke dynamics and stylometry features. As all messages are also labelled with gender, it is suitable for use in this project.

3.2 Dataset

The dataset consisted of 12 fields containing important data and metadata for each message. The fields and their descriptions are found in Table 3.1. Example dataset records are seen in Figure 3.2.

Field	Comment
Message	The content of the message.
Language	The language the chat was performed in. Viable options were Norwegian and English.
SenderID	An anonymous, randomized ID belonging to the sender of the message.
SenderGender	The gender of the person that sent the message.
SenderAge	The age of the person that sent the message.
ReceiverID	An anonymous, randomized ID belonging to the receiver of the message.
RecieverGender	The gender of the person that received the message.
ReceiverAge	The age of the person that received the message.
RoomID	A randomized ID for identifying the conversation.
UserAgent	Various info about the chatters' technical equipment used during the chat, such as web browser and operating system.
Timestamp	The time the message was sent.
KDinfo	Information about keystroke actions. See Table 3.2 for more details.

Table 3.1: The general structure of the records found in the dataset

Message	Language	SenderID	SenderGender	SenderAge	ReceiverID	ReceiverGender	ReceiverAge	RoomID	UserAgent	Timestamp	KDinfo
'hei!'	'norsk'	'16ZGF2BK'	'male '	23	'ZYKEZMDH'	'male '	62	'62d26aa2-5...	'Mozilla/5.0 (...	'2021-02-03T1...	*6x1 struct*
'heisann, hvordan står det til på en onsdag?'	'norsk'	'ZYKEZMDH'	'male '	62	'16ZGF2BK'	'male '	23	'62d26aa2-5...	'Mozilla/5.0 (...	'2021-02-03T1...	*57x1 struct*

Figure 3.2: Examples of records found in the dataset

The chat participants were not given any specific topics to talk about, but were instructed to speak freely and naturally. A consequence of this is that the chat participants might reveal personal information, such as name or location, during the course of the conversation. To avoid inclusion of personal information in the final dataset, any personal information apparent in the chat messages were removed by manual inspection. The pieces of personal information were then replaced with appropriate placeholder labels (names were replaced with "#NAME", locations were replaced with "#LOC" and URLs were replaced with "#URL").

The field KDinfo is more complex than the other fields, and therefore requires a discussion on its own. All other fields contain either simple integer values or character arrays, while KDinfo contains a struct[2] with several fields of its own. An example of a struct found in KDinfo is displayed in Figure 3.3. The fields of KDinfo is described in Table 3.2.

keyCode	key	TimeDown	TimeUp	RelTD	RelTU	Dur	LatRP	LatPP	LatRR	LatPR
72	'H'	'2021-02-03T1...	'2021-02-03...	3600207	3600289	82	NaN	NaN	41	123
16	'Shift'	'0000-00-00T0...	'2021-02-03...	NaN	3600330	NaN	158	NaN	245	NaN
69	'e'	'2021-02-03T1...	'2021-02-03...	3600488	3600575	87	44	131	124	211
73	'i'	'2021-02-03T1...	'2021-02-03...	3600619	3600699	80	554	634	640	720
49	'!'	'2021-02-03T1...	'2021-02-03...	3601253	3601339	86	NaN	NaN	63	149
16	'Shift'	'0000-00-00T0...	'2021-02-03...	NaN	3601402	NaN	NaN	NaN	NaN	NaN

Figure 3.3: Example of a struct stored in the field KDinfo

Field	Comment
keyCode	The ASCII code of the pressed key (case insensitive).
key	The value of the pressed key.
TimeDown	The time the key was pressed.
TimeUp	The time the key was released.
RelTD	Relative time value for when the key was pressed. Used to calculate LatPP, LatPR, LatRP and Dur.
RelTU	Relative time value for when the key was released. Used to calculate LatRR, LatPR, LatRP and Dur.
Dur	The total duration of the keystroke.
LatRP	Latency between release of the first key and press of the second key in a bigram.
LatPP	Latency between press of the first key and press of the second key in a bigram.
LatRR	Latency between release of the first key and release of the second key in a bigram.
LatPR	Latency between press of the first key and release of the second key in a bigram.

Table 3.2: The general structure of the struct KDinfo

The fields keyCode, key, TimeDown and TimeUp are collected by key-logging. The fields RelTD and RelTU are dervied from the timestamp in the particular record (see Figure 3.2 and Table 3.1), and represents relative values that can be used to calculate the remaining five fields. How the calculations are performed is discussed in the following paragraph. Recall from Section 2.1.2 the definition

[2]A data structure used in MATLAB. Similar to dictionaries in other programming languages.

of the features LatPP, LatPR, LatRP, LatRR and duration. Consider then a bigram AB, where A is the first key and B is the second key. Consider also that A_P and A_R denotes the press and release time for the first key, and B_P and B_R denotes the press and release time for the second key. Duration of the keys in a bigram can thus be calculated as $A_R - A_P$ and $B_R - B_P$. In KDinfo, this is achieved by calculating the difference between RelTD and RelTU for each keystroke.

LatRP of a bigram can be calculated by $B_P - A_R$. In KDinfo, this is achieved by calculating the difference between RelTU of the first key and RelTD of the second key.

LatPP of a bigram can be calculated by $B_P - A_P$. In KDinfo, this is achieved by calculating the difference between RelTD of the first key and RelTD of the second key.

LatPR of a bigram can be calculated by $B_R - A_P$. In KDinfo, this is achieved by calculating the difference between RelTD of the first key and RelTU of the second key.

LatRR of a bigram can be calculated by $B_R - A_R$. In KDinfo, this is achieved by calculating the difference between RelTU of the first key and RelTU of the second key.

These features can also be explained visually, as in Figure 3.4.

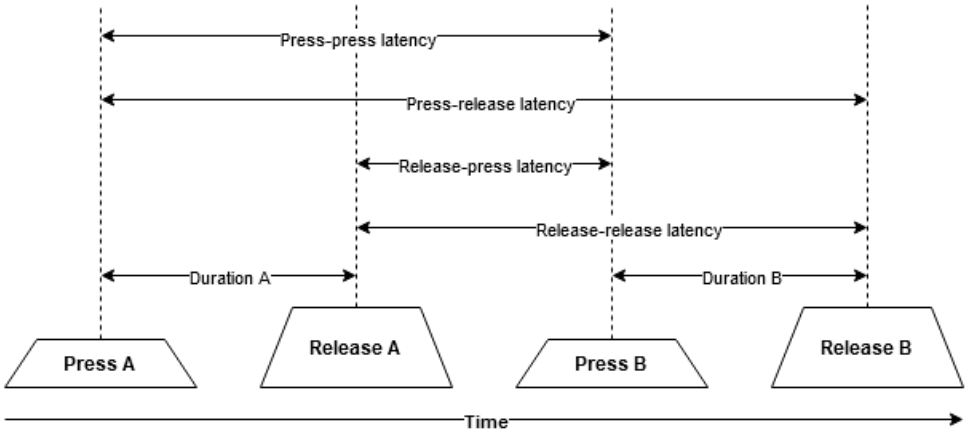

Figure 3.4: Visual explanation of common keystroke dynamics features

As can be seen in Figure 3.3, some of the fields have occurrences of NaN values. This is caused by an inability to calculate or collect those particular values. The last row will always contain NaN values for the fields LatRP, LatPP, LatPR and LatRR because these are bigram features which cannot be calculated without a second key following the first one. In addition, the UserAgent might not support registration of certain keystroke events. Some user agents might only support registration of key-down events, but not key-up events or vice versa. Some user agents might only support both key-down and key-up events for specific keys. In Figure 3.3, one can see that it was not possible to capture TimeDown for the Shift key, and as a result, the fields requiring this value is inhabited by NaN. Table 3.3

highlights some key numbers regarding the dataset, which serves the purpose of summarizing its properties.

Before performing any analysis, we removed all conversations participants with less than 5 written messages, as these instances were considered to not contain enough data. The updated properties of the dataset after removal of these instances can be seen in Table 3.4.

Property	Value (Norwegian / English)
Number of participants	64 / 18
Number of chat conversations	167 / 29
Number of messages	5898 / 647
Average number of messages per conversation	35.3 / 22.3
Average number of messages per person	92.2 / 35.9
Average number of keystrokes per person	6828.4 / 2352.4
Average number of keystrokes per message	74.1 / 65.4
Average number of characters per message	57.2 / 52.1
Male chat participants	16 / 8
Female chat participants	48 / 10
Male messages	1737 / 387
Female messages	4161 / 260
Average age of participants	32.5 / 33.1

Table 3.3: Properties of the full dataset

Property	Value (Norwegian / English)
Number of participants	57 / 13
Number of chat conversations	105 / 15
Number of messages	5719 / 614
Average number of messages per conversation	54.5 / 40.9
Average number of messages per person	100.3 / 47.2
Average number of keystrokes per person	7562.1 / 3198.1
Average number of keystrokes per message	75.4 / 67.7
Average number of characters per message	58.1 / 53.0
Male chat participants	15 / 6
Female chat participants	42 / 7
Male messages	1680 / 364
Female messages	4039 / 250
Average age of participants	32.5 / 33.8

Table 3.4: Properties of the dataset after deleting conversation participants with less than 5 written messages

Chapter 4

Data analysis

This chapter will describe the second part of our methodology, which is the data analysis. Key elements are which keystroke dynamics and stylometry features we utilized, and how we used these to perform gender classifications. The results of the data analysis will be discussed in Chapter 5. All software that was written to aid this data analysis, was written in MATLAB [48].

4.1 Feature extraction

Feature extraction is the process of extracting characteristics that can be used to distinguish two or more classes from each other, in this case between the two classes male and female. Feature extraction is a necessary prerequisite before any classification task, as the features serve as the data the classifier will base its decision upon. This section will describe the keystroke dynamics and stylometry features that were extracted. The extracted features are a combination of features that have been used in earlier research with promising results (see Chapter 2) and features that has not been widely studied earlier, but which we suspect could have an effect on distinguishing males and females.

4.1.1 Keystroke dynamics features

As described in Chapter 2, the keystroke dynamics features for gender classification tends to be a combination of LatPP, LatPR, LatRP, LatRR and durations. This has historically provided good accuracy in several environments. These features will thus also be included in this analysis. These features were easily extracted, as they were already provided in the dataset in the fields Dur, LatPP, LatPR, LatRP and LatRR. See Chapter 3 for more information regarding the dataset.

The features LatPP, LatPR, LatRP, LatRR were extracted for the 50 most frequently occurring bigrams in the dataset. Only the most frequent bigrams were used as a mean to reduce the total number of features. Considering the set of 95 printable ASCII characters, there would be a total of $95 \cdot 95 = 9025$ bigrams. With 4 features for each bigram (LatPP, LatPR, LatRP, LatRR), the total number

of bigram features would be $4 \cdot 9025 = 36100$. Most of these 9025 bigrams does however never/very seldomly appear in normal Norwegian chat conversations, neither by males nor females. Examples could be the bigrams "*:", "wx" and "|§". Figure 4.1 shows the count of the 500 most frequently appearing bigrams in the dataset. There was a total of 1314 unique bigrams appearing in the dataset, but the remaining 814 were excluded from the graph in Figure 4.1 due to readability. It can be seen in Figure 4.1 that there are some bigrams that are used considerably more often than others. The number 50 was selected as this includes all the top bigrams that appear significantly more frequently than the others, while still being low enough to not cause unnecessary computational expense. Increasing the number of features will require increased computational resources [49]. We also included one bigram that were not among the 50 most frequently used, namely "he". This bigram was number 58 sorted by frequency, but was added because almost all conversations started with this bigram due to its appearance in Norwegian greetings ("hei", "heisann" etc.). We therefore suspected this bigram could allow us to extract extra relevant information from the very first message in a conversation, which could prove beneficial for the sake of early gender detection. The complete list of the 51 selected bigrams can be seen in Appendix A.

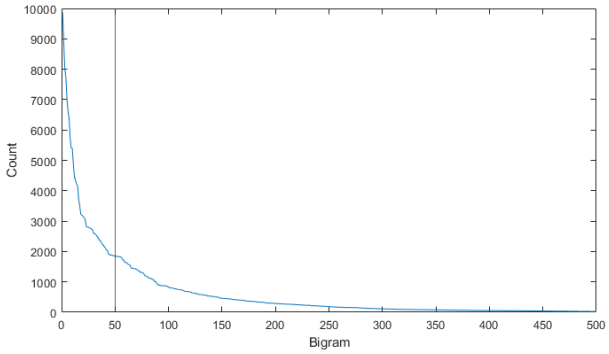

Figure 4.1: Count of the 500 most frequently appearing bigrams in the dataset

From each message, we then extracted all occurrences of LatPP, LatPR, LatRP and LatRR for the selected bigrams. We removed outliers by calculating the mean μ and standard deviation σ (see Equation (4.1) and Equation (4.2)) and then removed values that were more than 3 standard deviations away from the mean. We finally calculated a new mean based on the remaining values, which resulted in the final features.

$$\mu = \frac{\sum_{i=1}^{n} x_i}{n} \tag{4.1}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \mu)^2}{n-1}} \tag{4.2}$$

The duration features were extracted for all 29 Norwegian letters, numbers (0-9), some of the most common special characters (! . ? ; : , - + * ( ) /) and the spacebar. The Norwegian letters includes the English alphabet a-z, in addition to the Norwegian letters 'æ', 'ø' and 'å'. Other special characters were not included because there were very few instances of them found in the dataset, and they would thus not contribute much to distinguish males and females. From each message, we then extracted all durations of the selected characters and removed outliers using the same method as described in the previous paragraph. We also calculated new means based on the remaining values, which resulted in the final duration features. Despite being relatively frequently pressed, modifier keys (shift, ctrl, alt etc.) and backspace could not be included. This was because most participants' user agent did not allow recording of time down events of such keys, but only time up events. This made it impossible to calculate their durations. We did however extract the frequency of the most common modification keys (alt, shift, ctrl, tab, caps lock) and backspace. Even if timing information was not obtainable, it is still possible that the frequency of use might differ between the genders.

In addition, we extracted four features that we suspected could be of relevance for gender detection in a chat environment. The first one we called "hesitation", which is a measure of how long time it takes from the last keystroke in a message is released until the message is sent. This feature could display whether a person tends to read the message before sending (e.g., to search for typos) or if a person tends to immediately send the message after writing it. This feature can in other words tell something about the impulsiveness of a chat conversation participant. Extracting this feature consisted of calculating the difference between the field Timestamp and the last element in KDinfo.TimeUp for each message.

Another feature we extracted is "message time", which is the total time spent typing a message or alternatively the general typing speed of a conversation participant. Initial keystrokes that were not relevant for the message were ignored. An example of this is that for some users, the first registered keystrokes consisted of 'ctrl' + 'tab'. These keystrokes are obviously not related to the message, but was probably just used to select the correct tab in their web browser and should thus not be used to calculate typing speed. To extract this feature, we subtracted the time of the last key-release event with the first key-release event for each message. Key-release events were chosen because most chatters started messages with pressing the shift-key (to capitalize the first letter in a sentence), which caused many NaN values in the RelTD field (see Section 3.2 for more information).

The final two keystroke dynamics features we extracted were "space pause tendencies", which consists of a conversation participant's tendency to have longer pauses before or after pressing spacebar. This can show when in a sentence a chatter tends to take a "thinking break". We define a "pause" to be cases where the latency before or after a space is considerably larger than the other and lasts at least 500 milliseconds. We considered latencies lasting less than 500 milliseconds to be too short to be considered as "pauses". To extract these features, we did the following for each message:

Assume A is the key pressed before the spacebar, S is the spacebar press and B is the key pressed after the spacebar. $LatRP_{X,Y}$ denotes the LatRP value between keys X and Y. For each instance the spacebar was pressed, we calculated the formula:

$$r = \frac{-LatRP_{A,S} + LatRP_{S,B}}{LatRP_{A,S} + LatRP_{S,B}} \tag{4.3}$$

This returned a value r, where $-1 \leq r \leq 1$, which shows the difference in the ratio between $LatRP_{A,S}$ and $LatRP_{S,B}$.

$r < 0$ would imply that $LatRP_{A,S} > LatRP_{S,B}$, $r > 0$ would imply that $LatRP_{A,S} < LatRP_{S,B}$ and $r = 0$ would imply that $LatRP_{A,S} = LatRP_{S,B}$. We then defined "pauses" to be instances where $r < -0.5$ or $r > 0.5$, and the total latency is larger than 500 milliseconds. We chose to ignore cases where $-0.5 \leq r \leq 0.5$ because this would imply that the latency before and after a space is approximately equal and did most often occur when a message was written without any pauses. We then counted how many appearances of $r < -0.5$ and how many appearances $r > 0.5$ and divided each of them by the total number of spacebar presses. The features then consisted of these two values describing how often pauses appear before spaces and how often pauses appear after spaces.

A table summarizing all extracted keystroke dynamics features is found in Table 4.1.

Feature	Description
LatPP	Mean LatPP, with outliers removed, for the most frequent bigrams
LatPR	Mean LatPR, with outliers removed, for the most frequent bigrams
LatRP	Mean LatRP, with outliers removed, for the most frequent bigrams
LatRR	Mean LatRR, with outliers removed, for the most frequent bigrams
Durations	Mean duration, with outliers removed, for letters, numbers, spaces and the most common punctuation symbols
Modification and backspace frequency	The frequency of backspace and common modifier keys
Hesitation	Time between the last keystroke and send time of the message
Message time	Time between first and last keystroke in a message
Space pause tendencies	Tendency to have longer pauses before or after spaces

Table 4.1: Extracted keystroke dynamics features

4.1.2 Stylometry features

As seen in Section 2.2, the stylometry features that have been used for gender classification in earlier research is quite diverse. Where keystroke dynamics features almost always consists of a combination of LatPP, LatPR, LatRP, LatRR and durations, stylometry features varies from a wide selection of character-, word- and sentence-based features. There is however no consensus on which features perform the best. In this project we will mainly focus on relatively simplistic features on the character- and word-level. There are primarily two reasons for this.

1. In previously performed research, complex stylometry features have not necessarily increased performance (see Section 2.2).
2. The dataset is based on chat conversations, which one can assume tends to contain a rather simplistic language. This would make complex features difficult to extract.

All stylometry features were extracted from the field Message, which contains the content of the message. See Chapter 3 for more information regarding the dataset.

The first feature is average word length, which provides information regarding how long words a conversation participant tends to use. Calculating this consisted of summing all word lengths in each message and dividing the result on the number of words. In addition, we extracted the average lengths of sentences.

We also extracted the vocabulary richness, which is a feature that measures the complexity of a message. Vocabulary richness is a measure for how many different words a person tends to use. High vocabulary richness would thus imply that a conversation participant has a broad vocabulary and tends to use a relatively large amount of it in daily conversation.

Extracting this feature consisted of counting the number of unique words in each message. Another feature measuring message complexity is the frequency of long words. We extracted this by counting the number of words containing more than 10 characters and divided this by the total number of words. We also extracted the total character count of each message, which is a measure of message length.

Another category of features we used is the frequency of various characters. This includes letters (a-z, in addition to 'æ', 'ø' and 'å') and some common punctuation symbols (, . ! ? *). The character '*' is generally not often used in written text, but it is frequently used in chat environments to correct mistypings in earlier messages, which justified its inclusion here. These features might reveal much information about an author.

Character-densities can identify subtle differences in style and are, compared to word-densities, not as susceptible to outliers [50]. We also extracted the frequency of some of the most common emojis (:) ;) :-) ;-) :-(:P :D XD).

Calculating these features were done by counting each instance of the relevant characters/emojis and dividing them by the total number of characters. We also extracted the frequency of repeating punctuation symbols (... ?? !!). This was also

calculated by counting occurrences of each instance and dividing them by the total number of characters. A table summarizing all extracted stylometry features is found in Table 4.2.

Feature	Description
Average word length	Average length of words
Average sentence length	Average length of sentences
Vocabulary richness	Number of unique words in a message
Long words frequency	Frequency of words with more than 10 characters
Character count	Total number of characters in a message
Letter frequency	Frequency of each letter
Punctuation frequency	Frequency of certain punctuation symbols
Emoji frequency	Frequency of various emojis
Repeating punctuation	Frequency of sentences ending with two or more punctuation symbols

Table 4.2: Extracted stylometry features

4.2 Feature selection

Feature selection is the process of selecting a subset, out of all extracted features, that will be used to train the model. The purpose of this is to remove features that are not relevant for the classification task at hand. Removing irrelevant features can increase accuracy, decrease time complexity of training and make the model easier to understand, due to the lower number of features [49].

In this project, we used MATLAB's implementation of the Minimum Redundancy Maximum Relevance (MRMR) algorithm[1] for feature selection. MRMR is an algorithm that ranks features based on their relevance and redundancy based on calculating the mutual information between each pair of features and between each feature and the prediction variable (gender in this case). Relevant features are features that are able to make correct predictions, while redundant features are features that are highly correlated and thus provide the same information, which means that some of them can be removed without sacrificing performance. The algorithm assigns a score to each feature, where a high score implies high relevance and low redundancy, while a low score implies low relevance and high redundancy.

To remove the least useful features, we removed every feature with a score equal to 0 from the feature set. This allowed us to ignore the features that contained no relevant information.

[1] https://se.mathworks.com/help/stats/fscmrmr.html

4.3 Fusion

As described in Section 2.3, biometric fusion is the process of combining two or more biometric modalities with the goal of increasing accuracy. As this project uses the two modalities keystroke dynamics and stylometry, fusing is necessary to obtain the advantages of a multi-modal system. See Section 2.3 for more information about biometric fusion and previous work regarding the fusion of stylometry and keystroke dynamics.

In the setting of gender detection using keystroke dynamics and stylometry, there are primarily two methods of fusion that are relevant. These are feature-level and score-level fusion. There are no indications of which method will perform best in this setting, so both methods will be used and assessed. The results of each method of fusion will be discussed in Chapter 5.

4.3.1 Feature-level fusion

The first method of fusion we will use is feature-level fusion. This implies that the keystroke dynamics and stylometry features will be merged into an expanded feature set, before any classifications are done. This expanded feature set will then be used when training and testing the model.

To achieve this, we first wrote two different MATLAB-functions for extracting features for keystroke dynamics and stylometry respectively. The features were then stored in separate tables. We then merged the two tables by performing MATLAB's implementation of a join-operation,[2] which is a well-known method for combining tables. We also performed feature normalization by mapping each feature value to the range [0,1]. Feature normalization is necessary to remove differences in scale between the features, which can affect certain classifiers by giving some features increased weight [51]. A visualization of the feature-level fusion process is seen in Figure 4.2.

When performing a classification, in addition to the predicted label ("male" or "female"), the model will also return a classification score c which is a probability between 0 and 1 that describes the classifier's confidence in its decision. This classification score will be used in Section 5.2.3.

4.3.2 Score-level fusion

The second method of fusion is score-level fusion. Contrary to feature-level fusion, this consists of performing the fusion after the two classifications are performed. Two models, one for keystroke dynamics and one for stylometry, are created and are trained with keystroke dynamics and stylometry features respectively. When a classification is performed, each model generates a score which are then combined to determine the final classification score, which is then used to make the final classification decision.

[2] `https://se.mathworks.com/help/matlab/ref/table.join.html`

To achieve this, we used the same feature extraction functions mentioned in Section 4.3.1, but instead of joining the two feature sets, we used them on the two separate models. The features were normalized using the same method. After a message has been classified by both models, each model returns two scores, which are probabilities that a message is written by a male or female, according to the model. Score normalization was not needed as both models returned scores in the same domain (probabilities between 0 and 1).

We fused the scores from each modality by calculating combined probability scores for our two classes male and female. The male probability m was obtained by $m = P_{kd}(male) * w_{kd} + P_s(male) * w_s$, where $P_{kd}(male)$ and $P_s(male)$ are the probabilities that the message is male, according to the keystroke dynamics model and the stylometry model respectively, and w_{kd} and w_s are weights assigned to each modality, where $w_{kd} + w_s = 1$. As default, we used $w_{kd} = w_s = 0.5$. The female probability f would thus be $f = 1 - m$. The final classification would then be "male" if $m > f$ and "female" if $f > m$. If $m = f$, the model returns "undecided". The latter is however highly unlikely to occur. The final classification score c will be equal to m if the predicted label is "male", equal to f is the predicted label is "female" and 0.5 if the model returns "undecided". This classification score will be used in Section 5.2.3.

Alternatively, and possibly more intuitively, score-level fusion can be displayed visually, as in Figure 4.3.

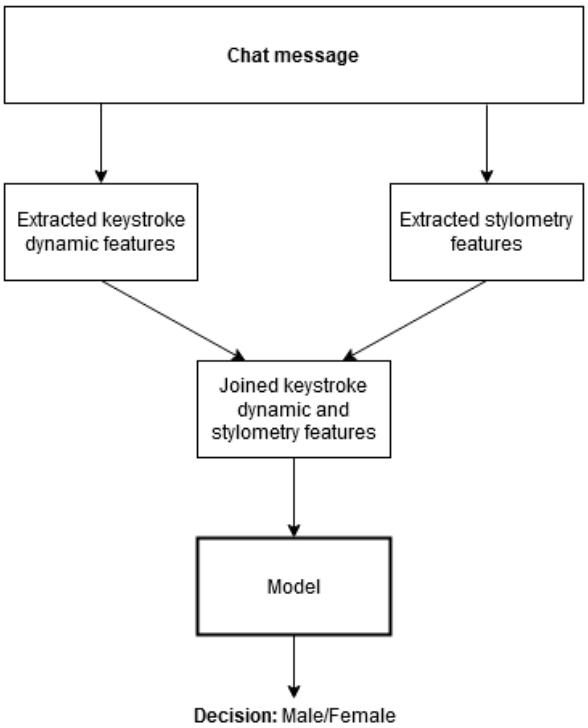

Figure 4.2: Visualization of feature-level fusion

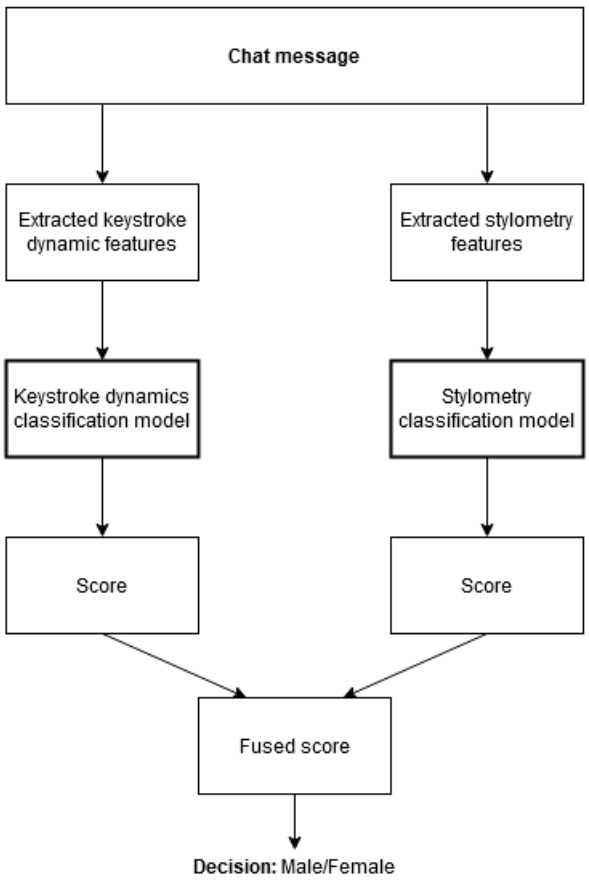

Figure 4.3: Visualization of score-level fusion

4.4 Classification

In general terms, classification is the process of assigning an observation to a particular category or class. In this project, the goal is to categorize a chat conversation participant into one of the two classes, male and female, based on his/her sent chat messages. As described in Chapter 2, this has most often been done by building machine learning models. Machine learning models will also be used in this project. This section will describe how we trained and tested these models.

4.4.1 Model training

For a machine learning model to make correct classifications, it first needs to be trained. This involves "feeding" the model with correctly labelled training data to make it able to recognize and correctly label unknown, unlabeled data. This is also known as supervised learning. The training data in this project consisted of conversations where we know the true gender (male or female) of the participants.

These conversations were obtained from the dataset described in Chapter 3. As seen in Table 3.3, the dataset contains way more messages written by females than by males. To balance the training data we therefore randomly removed messages written by females until the number of messages written by males and females were equal.

It is important to not use the same data for training and testing, and we thus split the dataset into separate sections to be used for training and testing respectively. To ensure that the split did not result in any unwanted consequences (e.g. the testing data only consisted of data that was easy to classify), we used k-fold cross-validation. k-fold cross-validation divides the dataset randomly into k sections, where $k-1$ sections are used for training and 1 section is used for testing. This process is iterated k times, where each iteration uses a different section for testing, and the remaining $k-1$ sections for training. We used $k = 5$.

We extracted the features described in Section 4.1 from the training data, and used these features to train the models. See Section 4.3 for how the training process differs between the two fusion methods. The training itself was performed by functions already included in MATLAB.[3] We trained models using k-NN, RF, SVM and NN, which have all been extensively used in earlier research (see Chapter 2). The reason for training several different models was to assess whether some perform better than others. A visualization of the training process can be seen in Figure 4.4.

4.4.2 Model testing

After a machine learning model has been trained, it needs to be tested. This is done to assess the performance of the model and check if it is able to make correct classifications. We can achieve this by providing the model with unlabelled data and checking how the predicted labels compare with the true labels.

To test the model, we first need testing data. As mentioned, this was obtained using k-fold cross-validation. After testing data has been obtained, we extracted features from each message in each conversation in the testing data and provided it as input to our trained models. The model then returns predicted label ("male" or "female") and classification score c of each message. The predicted labels and classification scores can then be used to determine the gender of the conversation participants. Further details regarding how we used these to perform early gender detection is described in Chapter 5. A visualization of the general model testing procedure is found in Figure 4.5.

[3] https://se.mathworks.com/help/stats/classification.html

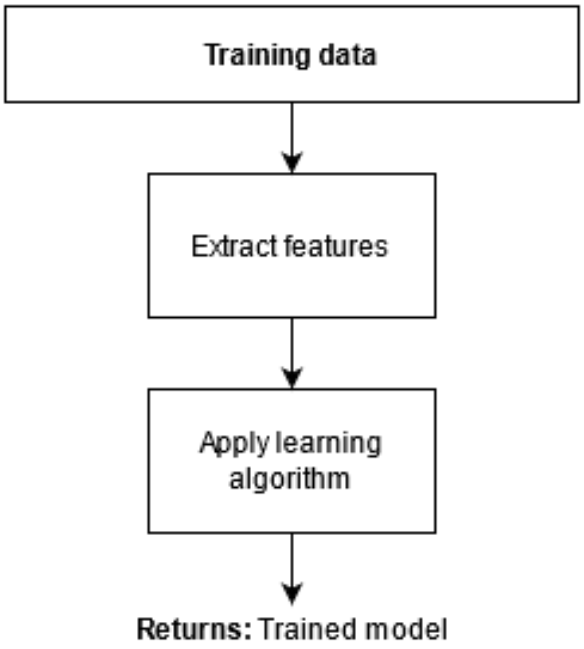

Figure 4.4: Visualization of model training

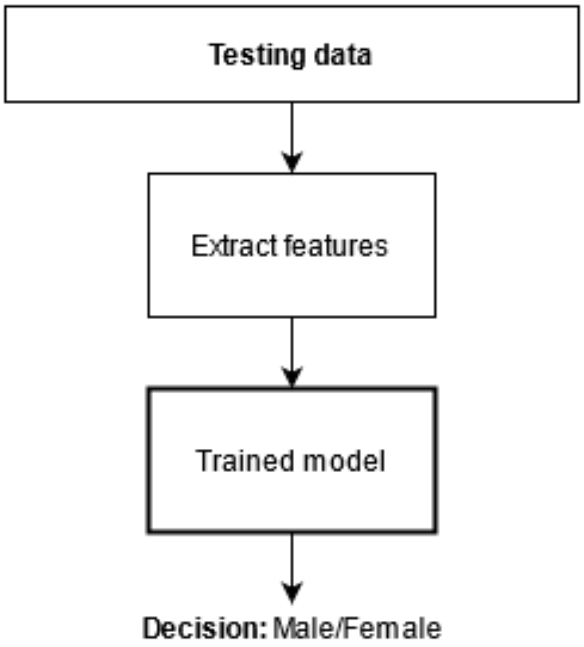

Figure 4.5: Visualization of model testing

Chapter 5

Results and discussion

This chapter will discuss the results obtained during data analysis. These results will be used to answer the research questions described in Section 1.5, where the main objective is to assess the possibility of early gender detection.

5.1 Baseline classification

Before trying to detect the gender of conversation participants early, we performed gender classifications based on the entire conversations. This allowed us to create a baseline that later can be used to assess the performance of the early classifications. Classifications based on all messages in a conversation are expected to be more accurate than classifications based on a lower number of messages. These baseline classifications can thus be used to determine how much accuracy is lost when making the classifications earlier. Table 5.1 shows the accuracy of our models, when basing the classification on entire conversations (average length of 28 messages per participant). To classify the gender of a conversation participant based on the entire conversation, we classified each sent message in the conversation separately, and the conversation would be considered correctly classified if the majority of the messages were classified as the correct gender. A conversation with an equal number of male and female classified messages would be considered not correctly classified. We also included accuracies when using each modality separately.

Based on the results in Table 5.1, one can observe some general tendencies. Firstly, one might notice that score-level fusion tends two perform slightly better than feature-level fusion. For both methods of fusion, the performance is nevertheless around what one could expect, based on the results we have found in related research (see Chapter 2). These accuracies are still a bit lower than the highest accuracies we have found in related research, but this is not crucial. In this project we are not primarily interested in obtaining the highest accuracy based on complete conversations, but rather how much the accuracy decreases when performing the classification earlier. One can also note that the RF and SVM classifiers performs better than the k-NN classifier. The RF classifier generally achieved the

Classifier	Fusion	Accuracy
RF	Feature	77%
k-NN	Feature	53%
SVM	Feature	76%
RF	Score	80%
k-NN	Score	67%
SVM	Score	75%
RF	KD	78%
k-NN	KD	70%
SVM	KD	77%
RF	Stylometry	70%
k-NN	Stylometry	51%
SVM	Stylometry	54%

Table 5.1: Performance of classifications based on entire conversations

best accuracies. A final observation is that classifications using solely keystroke dynamics seems to perform better than classifications using solely stylometry. This was also the case with the research in [16]. It is however worth noting that fusing the keystroke dynamics and stylometry scores improved the overall accuracy when using a RF classifier, which would imply that the stylometry features still contain relevant information, despite the relatively low accuracy when performing classifications based on stylometry features alone.

5.2 Early gender detection

This section will display and discuss our results when trying to perform the gender classification at an earlier stage in the conversations. As early gender detection has not yet been researched widely, we tried several approaches to assess whether some perform better than others.

5.2.1 General procedure

Even though we used different methods for early gender detection, they all followed the same general procedure. Each conversation in our test dataset, is processed message by message. Each participant in the conversation was assigned a default gender level of 0.5, which is updated after the processing of each message. See Section 2.4 for a more thorough description of gender levels. If a message is classified as male, the gender will increase and if classified as female, it will decrease. A gender level of 0 will thus imply complete certainty by the model that the conversation participant is male, while a gender level of 1 will in the same way imply that the participant is believed to be female. As the gender detection is to be performed early in a conversation, a certain stop criterion is needed to determine

when the final classification should be performed. One example of a stop criterion could be to perform the classification when the gender level approaches a certain value. To achieve as high performance as possible, we experimented with several different stop criteria.

5.2.2 Performance measures

When assessing the performance of our early gender detection scheme, we used two different metrics. The first one is accuracy loss, which is the difference in percentage points between the baseline classification accuracy (see Section 5.1) and the obtained accuracy when performing early gender detection. Ideally, the accuracy loss should be as low as possible, with 0 meaning that early gender detection is just as accurate as gender detection based on entire conversations. It is also possible that the accuracy loss is <0, which would in fact imply an accuracy gain. Our second performance metric is the average number of messages needed before the classification is made. Ideally, this value should also be as low as possible. A lower number of messages would result in the classification being performed earlier, but not necessarily with the same accuracy.

5.2.3 Gender level update mechanisms

In the general procedure we described in Section 5.2.1, there are two variables we can adjust. These are how the gender level is updated and how the stop criterion is defined. We will first discuss our three mechanisms for updating the gender level.

Static

The simplest update mechanism we used was with a static value. For each message, if it was classified as male, we incremented the gender level by a certain number s, and if classified as female, the gender level was decremented by the same value. We used $s = 0.025$.

Variable

One problem with the aforementioned solution, is that all messages are weighted equally. This is not optimal as long messages naturally contain more information than short messages. Longer messages thus gives the classifier more data to base the prediction upon. In addition, each classifier generates a classification score c for each message (see Section 4.3), where a higher classification score implies that the classification is more likely to be correct. Both these aspects argue that treating each message equal might not be the best solution. We therefore derived an equation that calculates a variable update value v, taking message length l (total number of characters, spaces included) and classification score c into account.

$$v = \frac{c\sqrt{l}}{100} \tag{5.1}$$

Multiplying c and l makes v grow as c and l increases, which means that the update value grows when the message is long and/or the classifier is confident in its decision. We used the square root of l to not assign too much weight to the message length and divided by 100 to obtain a sufficiently small number.

Hybrid

Our final update mechanism is a combination of static and variable updates. We used the same static value of $s = 0.025$, but we also introduced a confidence coefficient b which is used in combination with the static value to determine the gender level increment/decrement. The confidence coefficient can take the value of 0, 1, 2, or 4 and is a result of which bin the classification score is placed in. More precisely, given classification score c, the confidence coefficient b is 0 if $0.5 < c \leq 0.625$, 1 if $0.625 < c \leq 0.75$, 2 if $0.75 < c \leq 0.875$ and 4 if $0.875 < c \leq 1$. A higher confidence score results in a higher confidence coefficient. The final hybrid update value h is obtained by calculating $h = s \cdot b$. An advantage of this update mechanism is that it allows for not adjusting the gender level if the current message has a low confidence score. This could potentially affect performance.

5.2.4 Progression of gender levels

Before a stop criterion is determined, it can be interesting to observe how the gender level changes during the course of a conversation. This can aid us in finding the optimal stop criterion as we can notice at what time the gender level tends to become stable, if any.

We plotted a collection of graphs (Figures 5.1 to 5.6) were each line represents a conversation participant in our testing data. The graphs show the gender level at message number i in the conversation. The blue lines are instances where the true gender is male, and the red lines are instances where the true gender is female. The dots display at what time the conversation ended and the horizontal dashed line at 0.5 splits the gender level range into male and female sub-ranges. As the RF classifier achieved the best performance (by a slight margin) in the baseline classification, the graphs in this section will focus on gender level progressions when using this classifier.

See Appendix B for the gender level progressions when using all classifiers (RF, k-NN and SVM).

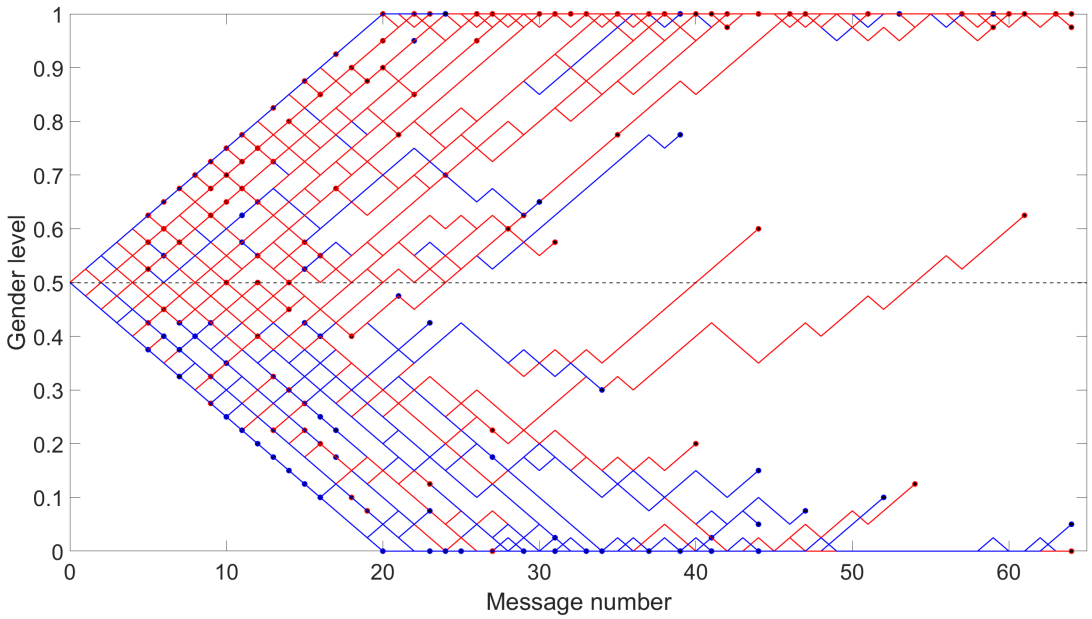

Figure 5.1: Gender level progressions using static gender level update mechanism and score-level fusion with the RF classifier

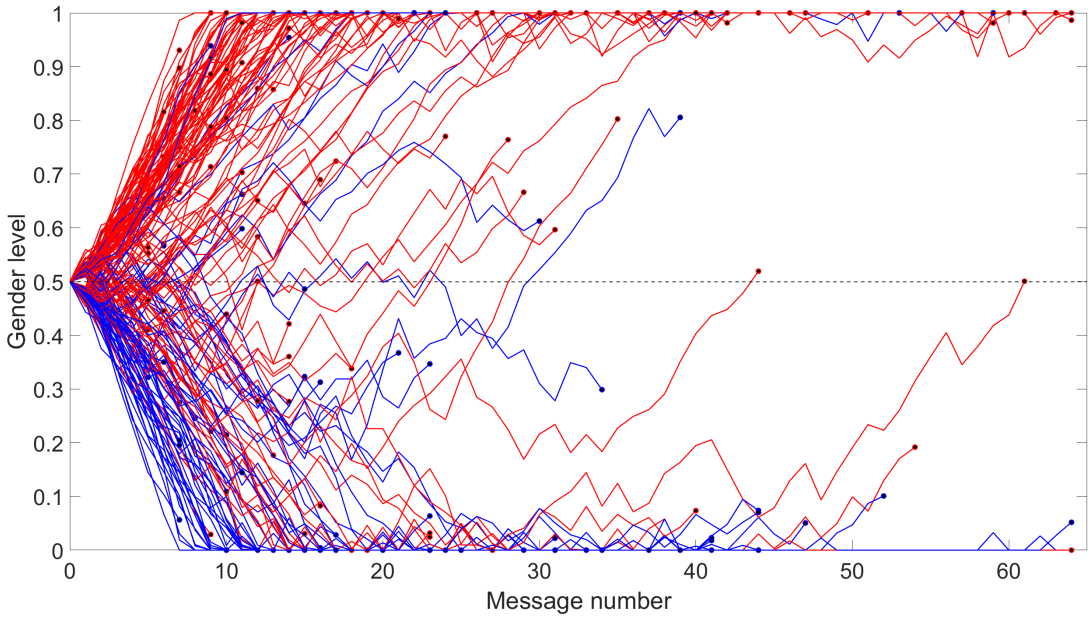

Figure 5.2: Gender level progressions using variable gender level update mechanism and score-level fusion with the RF classifier

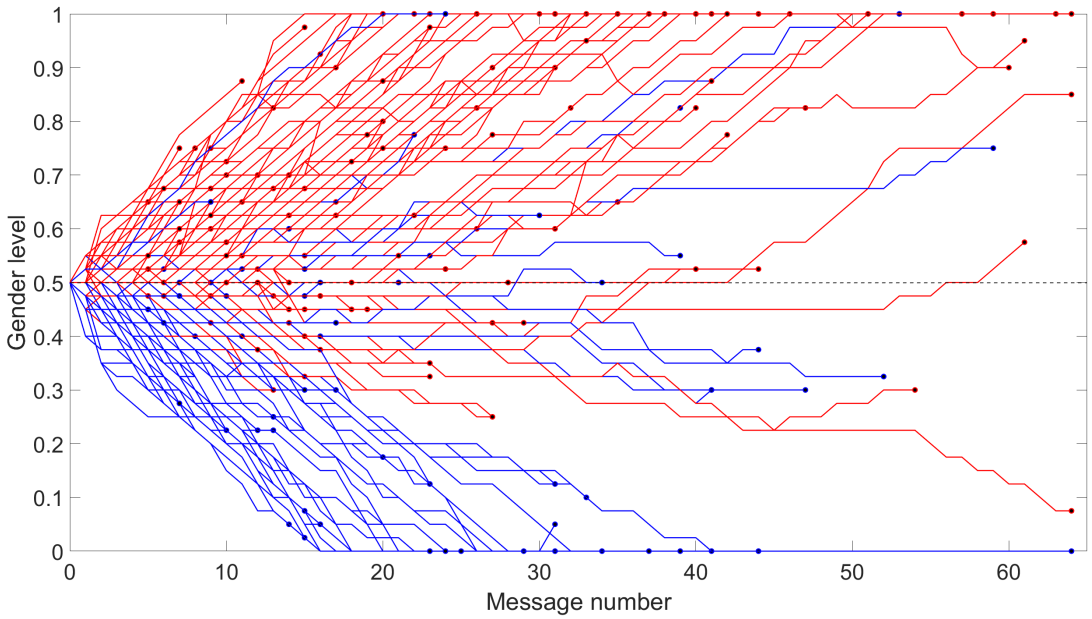

Figure 5.3: Gender level progressions using hybrid gender level update mechanism and score-level fusion with the RF classifier

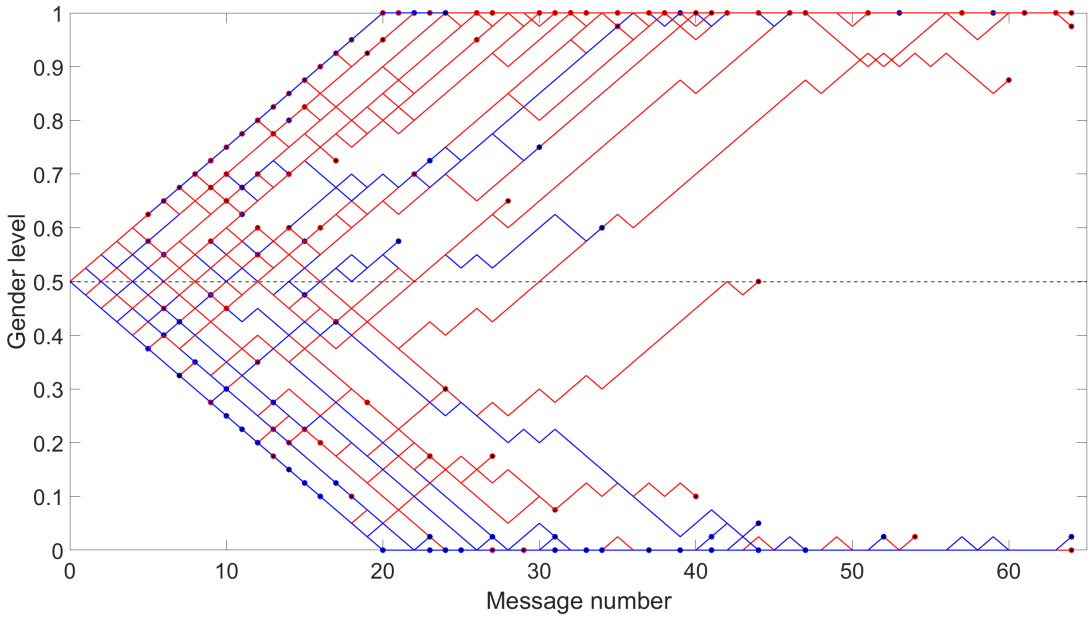

Figure 5.4: Gender level progressions using static gender level update mechanism and feature-level fusion with the RF classifier

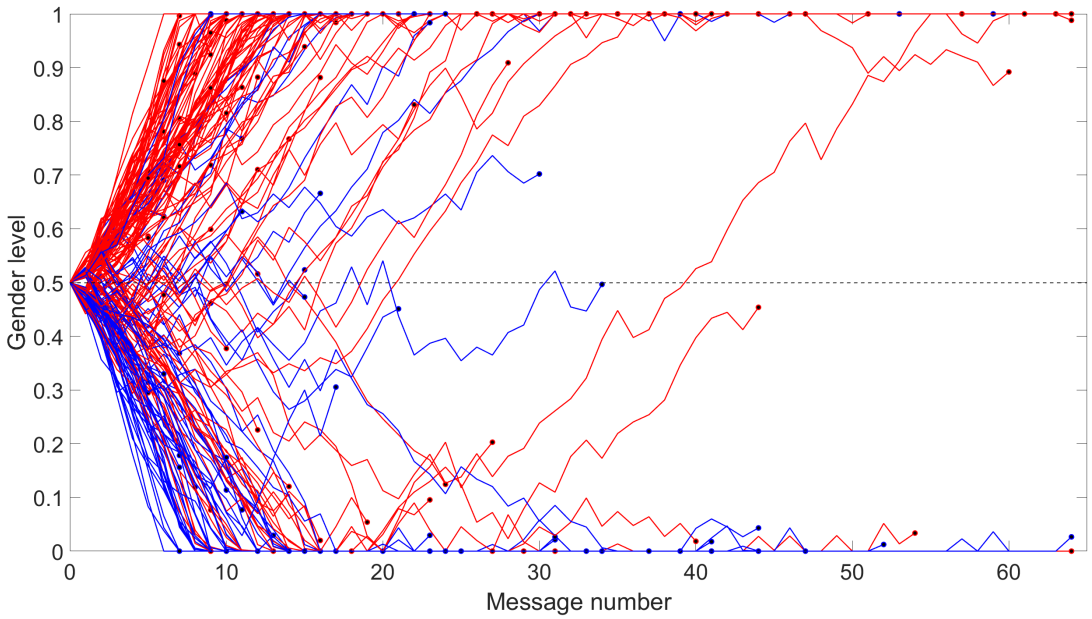

Figure 5.5: Gender level progressions using variable gender level update mechanism and feature-level fusion with the RF classifier

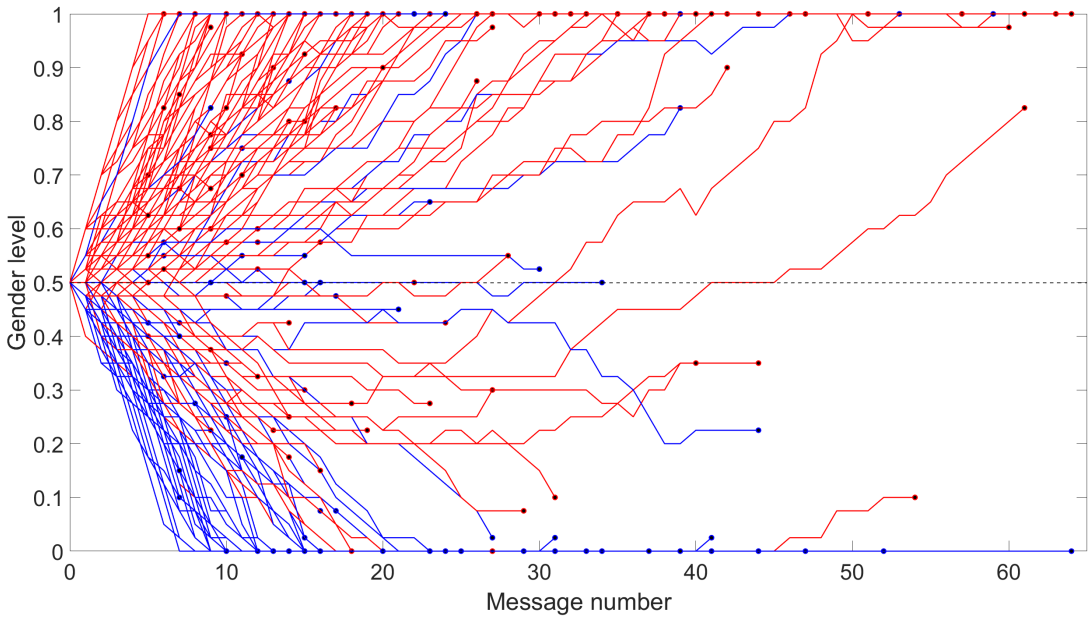

Figure 5.6: Gender level progressions using hybrid gender level update mechanism and feature-level fusion with the RF classifier

Based on Figures 5.1 to 5.6, one could categorize the conversation participants into three distinct categories. The first category consists of the participants whose gender level converges straight towards the correct gender. These are the male

participants whose gender level converges straight towards 0 and female participants whose gender level converges straight towards 1.

The second category consists of the participants whose gender level converges straight towards the false gender; females converging straight towards 0 and males converging straight towards 1. These two categories can be ignored when selecting a stop criterion for early gender detection. Because the gender level only moves in one direction, the result of the classification will not change no matter when the classification is performed.

The third category consists of the conversation participants whose gender level does not converge to either 0 or 1. Most of these participants have a gender level which, on different times, can be either above or below the default value of 0.5. These are the participants that can be affected by our stop criterion.

The first thing one might notice is that most conversation participants' gender level tend to converge straight towards 0 or 1 rather quickly. This means that only a few participants have a gender level that changes widely in both directions. This is highly positive for the case of early gender detection. If the gender level tends to only change in one direction, there will be no accuracy loss by performing the classification earlier than at the end of the conversation.

One can also notice that using a static or hybrid update mechanism increases the risk of having "undecided" classifications, which occur when the final gender level is exactly 0.5. This is however not a major issue as it only applies to a small number of the conversation participants and the risk decreases as the number of messages increases. Finally, one can see that all update mechanisms produce similar looking gender level progressions in regard to structure. All fit with one of the three categories we defined above. The only significant difference is that the variable update mechanism causes the gender level to converge to either 0 or 1 much faster than the two others. We did not observe any significant differences between the two methods of fusion.

Table 5.2 shows the accuracy obtained with the RF classifier when using different fusion methods and update mechanisms based on the gender level at the end of each conversation. The letters 'S', 'V' and 'H' represent our static, variable and hybrid gender level update mechanisms. Note in Table 5.2 that when using feature-level fusion, the variable gender level update mechanism performs better than the baseline accuracy. This is a promising sign of its proficiency as an update mechanism.

Fusion	Classifier	Male accuracy			Female accuracy			Total accuracy		
		S	V	H	S	V	H	S	V	H
Feature	RF	74%	76%	72%	79%	79%	77%	77%	78%	75%
Score	RF	79%	81%	69%	80%	80%	78%	80%	80%	75%

Table 5.2: End of conversation accuracies using different update mechanisms and methods of fusion

5.2.5 Absolute thresholds

The first stop criterion we tested, was with absolute thresholds. By "absolute", we mean that the final gender classification occurs right when one of the defined thresholds are reached. We tested several different threshold options, which allowed us to observe how the different performance measures were affected by different thresholds. This can be seen in Table 5.3. In this table, the two columns "Accuracy loss" and "Average number of messages" corresponds to our two performance measures. The letters 'S', 'V' and 'H' still represent our static, variable and hybrid gender level update mechanisms. Note that some of the configurations have a negative accuracy loss (or accuracy gain). This would imply that the accuracy is higher than what we achieved in the baseline classification. Negative accuracy loss only appears when using the variable or hybrid update mechanism, which means that these update mechanisms sometimes performs better than the simplistic approach we used in the baseline classification where a conversation participant was simply classified to be the gender which the majority of the messages were classified as. Recall also that when basing the classifications on entire conversations, the average number of messages were 28.

Based on Table 5.3, it is apparent that early gender detection is very well possible. In the most extreme scenario, where the gender of a conversation participant is only based on the first message, the accuracy loss is never larger than 10 percentage points (except when using the k-NN classifier with score-level fusion). The generally low accuracy loss confirms what we saw in Section 5.2.4, where most conversation participants' gender level converge straight towards the correct end of the gender level scale. The general trend in Table 5.3 is nevertheless that the lowest accuracy loss appear when using thresholds 0.0/1.0. As the thresholds moves towards 0.5, we can see that the accuracy loss increases, and the average number of messages decreases. This aligns well with what we expected. When basing the classification on less data, it is not surprising that the accuracy will suffer to some degree.

It is also interesting to see that our gender level update mechanisms affect performance in different ways. In this case, the best performance is achieved by our variable update mechanism. The variable update mechanism achieves, in most cases, the lowest accuracy loss, and in all cases, the lowest average number of messages. This update mechanism is the only one that takes message length into account and its performance shows that assigning heavier weight to longer messages has a positive effect on accuracy. Increasing the weight of longer messages also means that a single message can have a large impact on the gender level, which means that the gender level will approach the thresholds faster than with other update mechanisms.

Thresholds	Fusion	Classifier	Accuracy loss			Average number of messages		
			S	V	H	S	V	H
0.0 / 1.0	Feature	RF	0	-1	2	18	13	16
		k-NN	0	-4	-5	22	16	21
		SVM	0	-1	1	20	14	16
0.1 / 0.9	Feature	RF	0	0	2	16	11	14
		k-NN	0	-4	-5	20	14	19
		SVM	0	-1	1	19	13	16
0.2 / 0.8	Feature	RF	0	0	2	13	9	12
		k-NN	0	-3	-4	17	12	17
		SVM	0	-1	1	16	11	13
0.3 / 0.7	Feature	RF	0	0	2	10	7	9
		k-NN	0	0	-3	13	9	14
		SVM	0	-1	1	12	8	11
0.4 / 0.6	Feature	RF	1	3	3	6	5	6
		k-NN	2	-1	-3	7	5	9
		SVM	0	1	3	7	5	8
0.49 / 0.51	Feature	RF	8	8	4	1	1	2
		k-NN	-5	-3	0	1	1	2
		SVM	6	6	4	1	1	2
0.0 / 1.0	Score	RF	0	0	5	19	14	22
		k-NN	0	-6	3	23	18	25
		SVM	0	0	4	20	15	22
0.1 / 0.9	Score	RF	0	1	5	17	12	20
		k-NN	0	-6	3	21	16	24
		SVM	0	0	4	18	14	21
0.2 / 0.8	Score	RF	1	1	5	14	10	18
		k-NN	0	-6	3	19	14	23
		SVM	0	0	4	15	11	19
0.3 / 0.7	Score	RF	1	2	5	11	8	15
		k-NN	1	-4	3	15	10	21
		SVM	0	0	4	12	9	17
0.4 / 0.6	Score	RF	3	2	6	6	5	10
		k-NN	3	-1	4	8	6	16
		SVM	0	2	4	7	5	12
0.49 / 0.51	Score	RF	10	9	6	1	1	3
		k-NN	24	21	4	1	1	7
		SVM	2	2	9	1	1	5

Table 5.3: Performance of early gender detection with absolute thresholds

Our hybrid update mechanism also has some interesting properties. It generally needs relatively many messages to reach a threshold and the accuracy loss is generally not as low as with our variable update mechanism, but the accuracy loss is not widely affected by changing the threshold. By changing the thresholds from 0.0/1.0 to 0.49/0.51, the accuracy loss increases on average by 7.5 and 9.2 percentage points for static and variable update mechanism respectively, while it only increases by 2.9 percentage points on average for our hybrid update mechanism. This can most likely be attributed to the fact that this is the only update mechanism that allows for not updating the gender level if the confidence score is too low.

There is less to say about our static gender level update mechanism. It does not perform best on any of our performance measures, but the performance is by no means bad. Using feature-level fusion, it for example achieved no performance loss and more than halved the average number of messages when using thresholds 0.3/0.7.

The classifiers also differ in their performance. When it comes to accuracy loss, k-NN actually achieved the best performance in most cases. It did however always need the highest number of messages. It should also be noted that k-NN achieved the lowest baseline accuracy, which reduces the gain from the low accuracy loss. A higher baseline accuracy, as we achieved with RF and SVM, naturally leads to a higher tolerance for eventual accuracy loss. There is however one case where the accuracy loss of k-NN is much higher than the other classifiers. This occurred when using score-level fusion with thresholds 0.49/0.51. In this case, the accuracy loss when using a static or variable update mechanism was above 20 percentage points. This would imply that our k-NN classifier is more unreliable when using only the first message in a conversation, compared to our RF and SVM classifiers.

The RF and SVM classifiers were pretty similar in performance. RF needed a slightly lower average number of messages to reach the thresholds, while SVM achieved slightly lower accuracy loss. Based on Table 5.3, these classifiers should nevertheless both be considered to be better than k-NN due to the higher baseline accuracies. A slight edge could potentially be assigned to the RF classifier as this achieved the highest baseline accuracy.

The two methods of fusion do also not have any significant differences regarding performance. Both the accuracy loss and the average number of messages is approximately the same in all cases. The one exception is the case with k-NN using thresholds 0.49/0.51 with static or variable update mechanism, as discussed above. In this case, the accuracy loss was way higher when using score-level fusion. This is however not a particularly important difference. When the classification is based only on the first message, one should accept that accuracy is lower than classifications based on more messages. In a real-world scenario one should thus not choose thresholds that close to 0.5. In addition, one can see that k-NN achieved negative accuracy loss when using feature-level fusion with the same thresholds, which also shows that thresholds that close to 0.5 are obviously unstable at best.

Based on our findings in Table 5.3, we could try to determine an optimal set of thresholds. This is however difficult as our two performances measures are somewhat conflicting. Thresholds with the lowest accuracy loss often need the highest average number of messages and vice versa. How a compromise should be made would differ from application to application based on one whether quickness or correctness is of most importance. If one were to select thresholds representing a middle ground, thresholds between 0.2/0.8 and 0.3/0.7 should be appropriate. Based on our findings in Table 5.3, this would ensure virtually no accuracy loss and approximately halving the average number of messages from 28 to around 14 (some minor variations based on method of fusion, classifier and update mechanism). If some accuracy loss can be tolerated, the thresholds could be set to 0.4/0.6, which would result in an average number of messages around 5, while still keeping the accuracy loss <5%.

5.2.6 Introducing stability thresholds

In the previous section, we described the stop criterion where the final classification was made once the conversation participant's gender level reached a defined threshold. It was not taken into account what happens after this point in the conversation. We therefore also tested another stop criterion where the classification was performed when the gender level has been above a certain threshold for N consecutive messages. In our case, we used $N = 3$. As can be seen, in for instance Figure 5.2, there are several conversation participants whose gender level first dips quite far below 0.5 before changing direction and stabilizes above 0.5. Not making a decision before one has stayed for a certain time above/below a certain threshold could have an effect on performance. Table 5.4 displays performance with this new stop criterion.

As seen in Table 5.4, the general trend is that stability thresholds achieve slightly lower accuracy loss, while also needing a slightly higher average number of messages. The differences are however too small to be considered significant. This does however teach us one important thing: Once a gender level threshold is reached, the gender level does normally not return to the other side. This discovery is highly in favour of early gender detection because it tells us that when a gender level threshold is reached, there is not much to gain by waiting and see whether the gender level will stay there; in most cases it does.

It is therefore difficult to claim that stability thresholds are better than absolute thresholds. First of all, whether stability thresholds are the better option would be situation dependent and would vary based on whether accuracy loss or a low number of messages is prioritized. In a situation where minimizing accuracy loss would have top priority could be interpreted as an argument in favour of stability thresholds, but one must also remember that the attributes of stability thresholds, namely slightly lower accuracy loss and a slightly higher number of messages, could however also be achieved by adjusting the absolute thresholds.

Thresholds	Fusion	Classifier	Accuracy loss			Average number of messages		
			S	V	H	S	V	H
0.0 / 1.0	Feature	RF	0	-1	2	19	14	17
		k-NN	0	-4	-5	23	17	22
		SVM	0	-1	1	21	16	17
0.1 / 0.9	Feature	RF	0	-1	2	17	13	15
		k-NN	-1	-4	-5	21	16	20
		SVM	0	-1	1	19	14	16
0.2 / 0.8	Feature	RF	0	0	2	15	11	14
		k-NN	0	-3	-4	18	14	18
		SVM	0	-1	1	17	13	14
0.3 / 0.7	Feature	RF	0	0	2	12	9	11
		k-NN	0	-2	-3	14	11	16
		SVM	0	-1	1	14	11	12
0.4 / 0.6	Feature	RF	2	0	3	8	7	8
		k-NN	2	-1	-2	10	8	11
		SVM	0	1	2	10	8	9
0.49 / 0.51	Feature	RF	2	4	3	4	3	4
		k-NN	1	-1	-3	4	3	5
		SVM	2	5	3	4	4	4
0.0 / 1.0	Score	RF	0	0	5	20	16	22
		k-NN	0	-6	3	23	19	26
		SVM	0	0	4	21	17	23
0.1 / 0.9	Score	RF	0	0	5	18	14	21
		k-NN	0	-6	3	22	18	25
		SVM	0	0	4	19	15	21
0.2 / 0.8	Score	RF	0	1	5	16	12	19
		k-NN	0	-6	3	20	16	23
		SVM	0	0	4	16	13	20
0.3 / 0.7	Score	RF	1	1	5	12	10	16
		k-NN	1	-5	3	16	13	21
		SVM	0	0	4	14	11	18
0.4 / 0.6	Score	RF	3	3	6	8	8	12
		k-NN	0	-2	4	11	9	17
		SVM	0	1	4	9	8	14
0.49 / 0.51	Score	RF	3	5	6	4	4	5
		k-NN	6	4	3	4	4	9
		SVM	2	3	8	4	3	7

Table 5.4: Performance of early gender detection with stability thresholds

We saw in Section 5.2.5 that as the thresholds moved close towards 0.0/1.0, the accuracy loss was lowered and the average number of messages increased. There is therefore, in our case, not much to gain by introducing stability thresholds. One should however not completely discard such thresholds either. Compared to our findings in Chapter 2, our dataset is rather small so we should not over-generalize the observations, and it is also possible that there exist other cases where stability thresholds would be a definite improvement. More research is needed to draw a final conclusion.

5.2.7 Separating keystroke dynamics and stylometry

We have in the previous sections looked at how keystroke dynamics and stylometry have performed in combination. It can also be interesting to look at each modality separately. This allows us to discover how each modality affects the gender classification. We saw in Section 5.1 that keystroke dynamics generally performs better than stylometry in regards to accuracy at the end of conversation, but there are still aspects that the baseline accuracy cannot uncover. One example is how the gender levels progresses throughout the conversations. We therefore plotted the gender level progressions for each modality separately when using the RF classifier, which can be seen in Figures 5.7 to 5.12. We will only focus on the RF classifier in this section, as it generally achieved the best performance. Table 5.5 shows relevant accuracies associated with the gender level progressions in Figures 5.7 to 5.12.

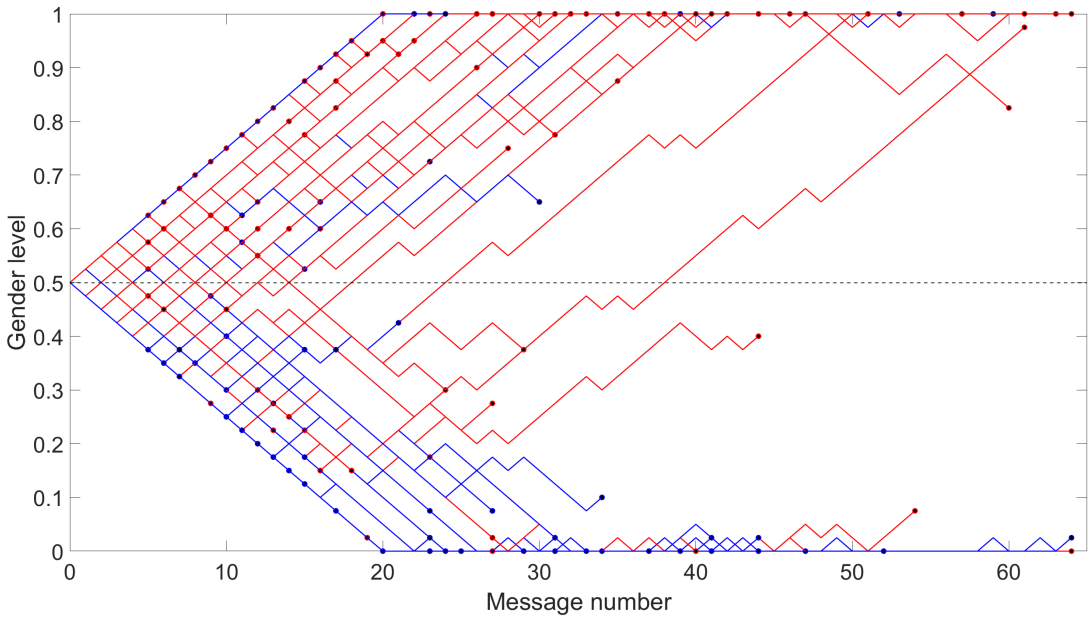

Figure 5.7: Keystroke dynamics gender level progressions when using static gender level update mechanism with the RF classifier

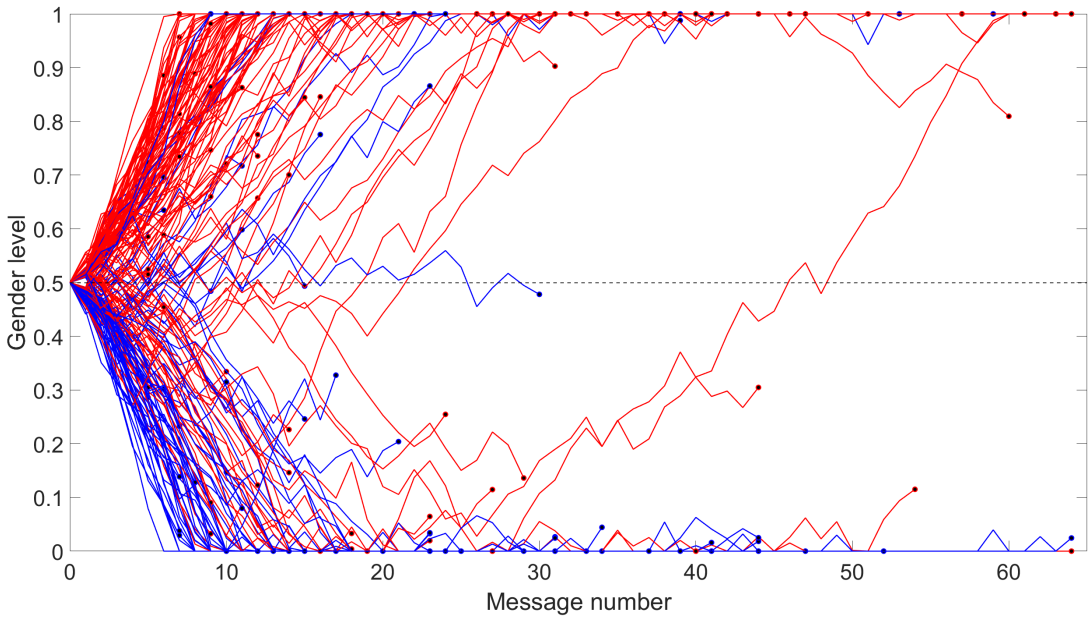

Figure 5.8: Keystroke dynamics gender level progressions when using variable gender level update mechanism with the RF classifier

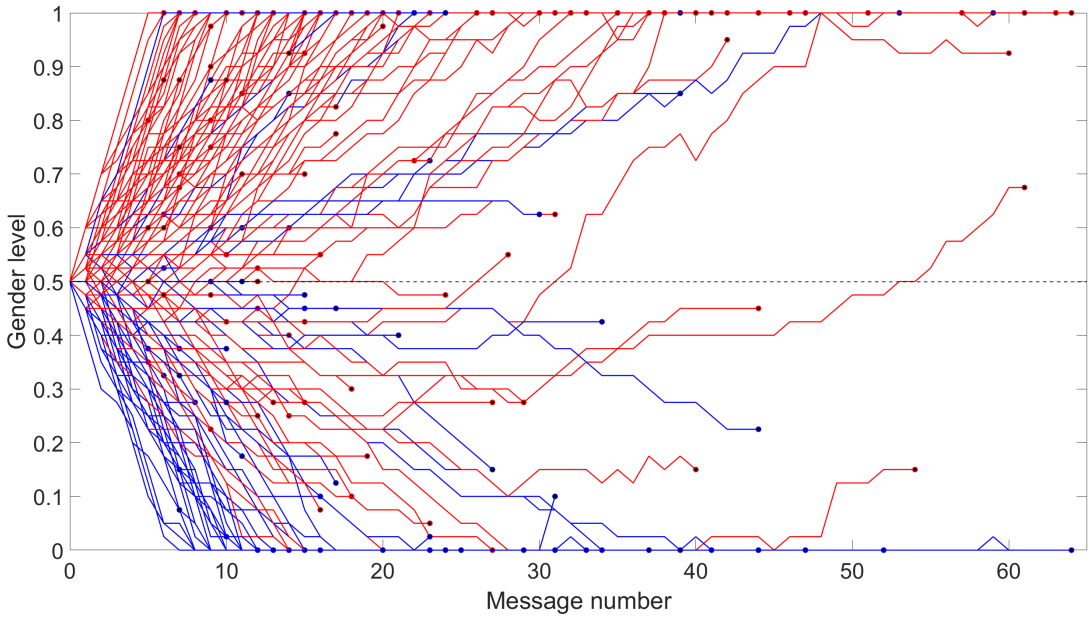

Figure 5.9: Keystroke dynamics gender level progressions when using hybrid gender level update mechanism with the RF classifier

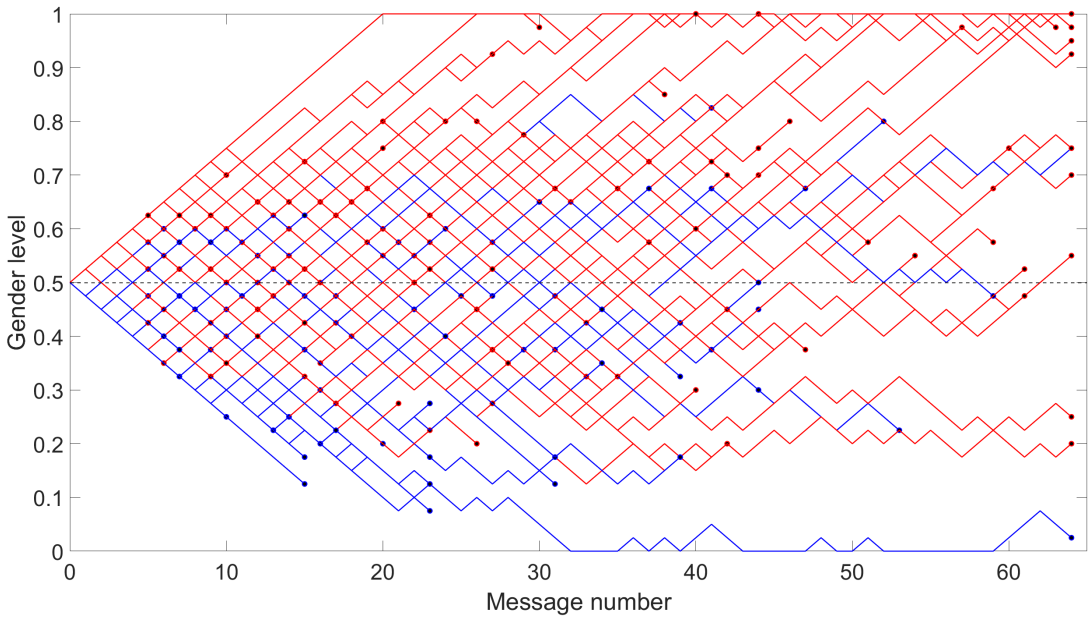

Figure 5.10: Stylometry gender level progressions when using static gender level update mechanism with the RF classifier

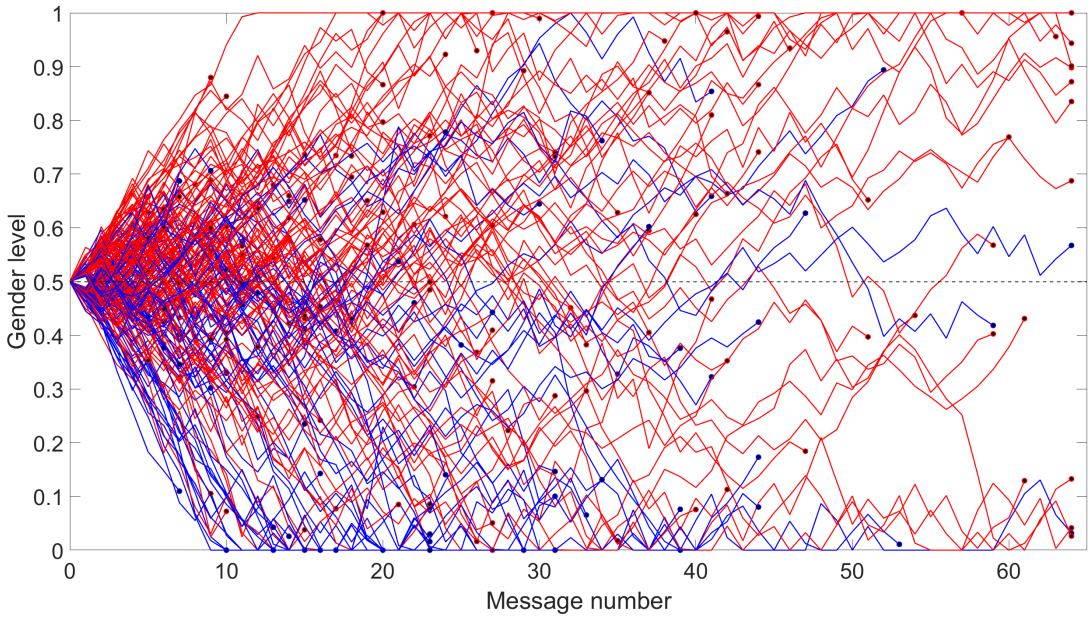

Figure 5.11: Stylometry gender level progressions when using variable gender level update mechanism with the RF classifier

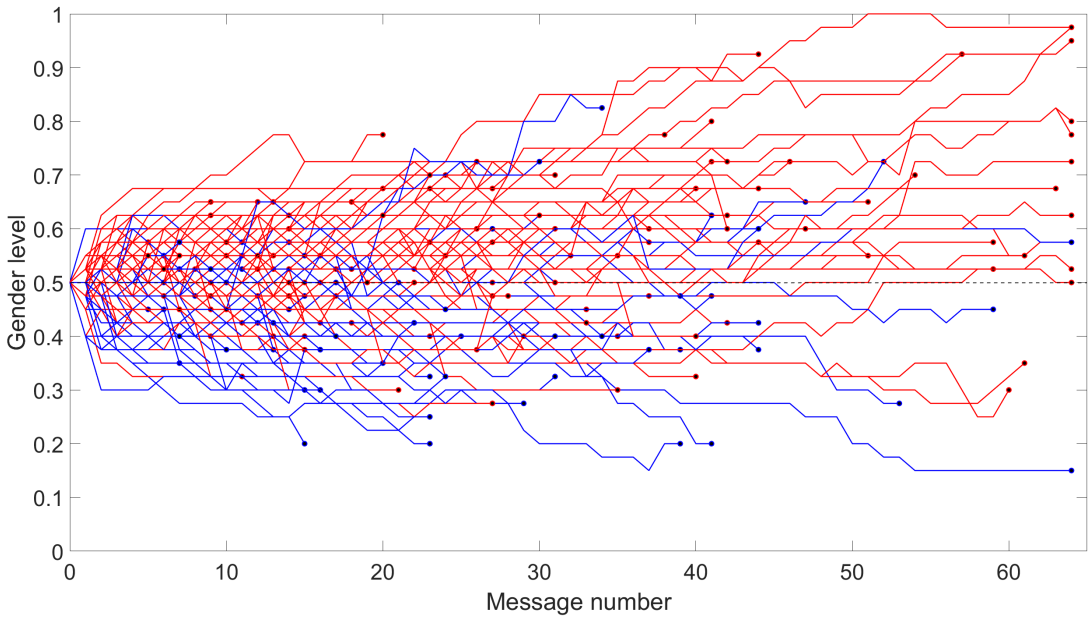

Figure 5.12: Stylometry gender level progressions when using hybrid gender level update mechanism with the RF classifier

Modality	Classifier	Male accuracy			Female accuracy			Total accuracy		
		S	V	H	S	V	H	S	V	H
KD	RF	76%	79%	76%	79%	79%	76%	78%	79%	76%
Stylometry	RF	72%	75%	65%	69%	57%	63%	70%	63%	64%

Table 5.5: End of conversation accuracies using separate modalities with different update mechanisms

Based on Figures 5.7 to 5.12, the most significant observation is that stylometry gender level progressions are way more cluttered than with keystroke dynamics. By "cluttered" we mean that there are many conversation participants whose gender level tend to stay relatively close to the gender separation value of 0.5 and often swings both above and below this value. Visually, it is also difficult to manually notice any distinct patterns. This is most apparent in Figure 5.11 and Figure 5.12. By looking at the stylometry gender level progressions using a hybrid update mechanism (Figure 5.12), one can also see that this graph is flatter than most others, due to the fact that there are many cases where the gender level does not decrease or increase. This means that there are many messages where the stylometry classifier has a low confidence in its classification. Basing a system for early gender detection on solely stylometry appears therefore to be somewhat challenging, at least in our case. It is however important to remember that when using the RF classifier, we achieved a higher baseline accuracy when including stylometry in addition to keystroke dynamics, which shows that stylometry is still useful in regard to gender detection.

Contrary to stylometry, the gender level progressions using only keystroke dynamics are mostly similar to the progressions displayed in Section 5.2.4 and there are thus not any additional observations to be made.

We also tested combining the gender level progressions from the two modalities into a single one. This must not be confused with our previously used fusion methods where the modalities are combined into a single value which is used to update the gender level, as described in Section 4.3. In this case, we first obtain the gender level update value (static, variable or hybrid) using only keystroke dynamics. This value is then used to update the gender level. We then, from the same message, obtain the gender level update value (static, variable or hybrid) using only stylometry. This second value is then used to update the gender level again. We also divided each gender level update value by 2 to ensure updates with the same proportions as when we used traditional feature-level and score-level fusion. These gender level progressions are displayed in Figures 5.13 to 5.15, and Table 5.6 shows relevant accuracies associated with the gender level progressions in these figures.

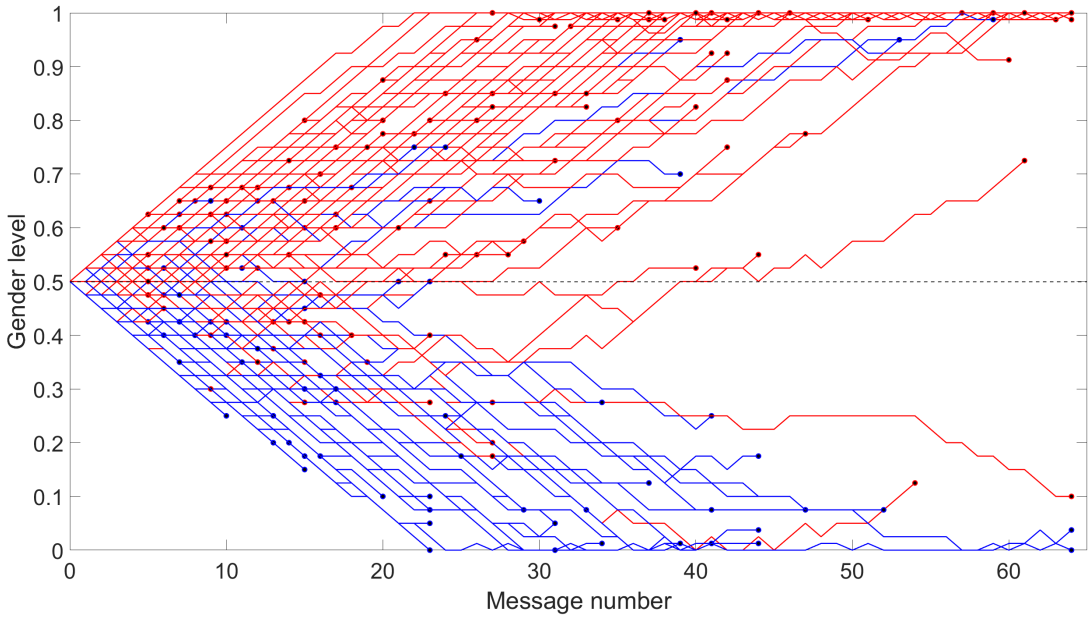

Figure 5.13: Gender level progressions using static two-step gender level update mechanism with an RF classifier

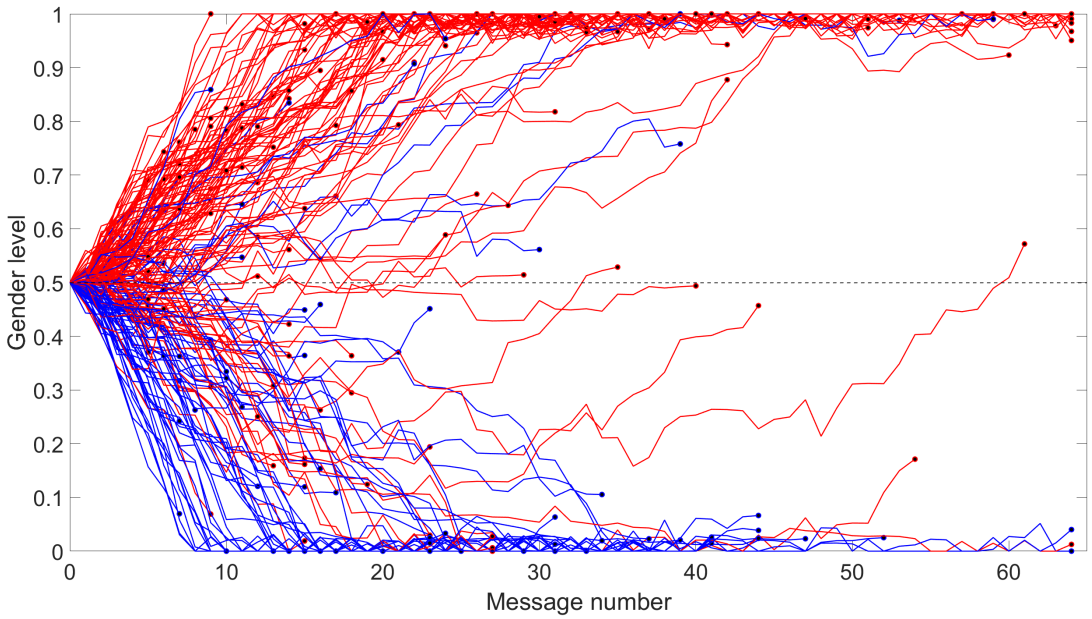

Figure 5.14: Gender level progressions using variable two-step gender level update mechanism with an RF classifier

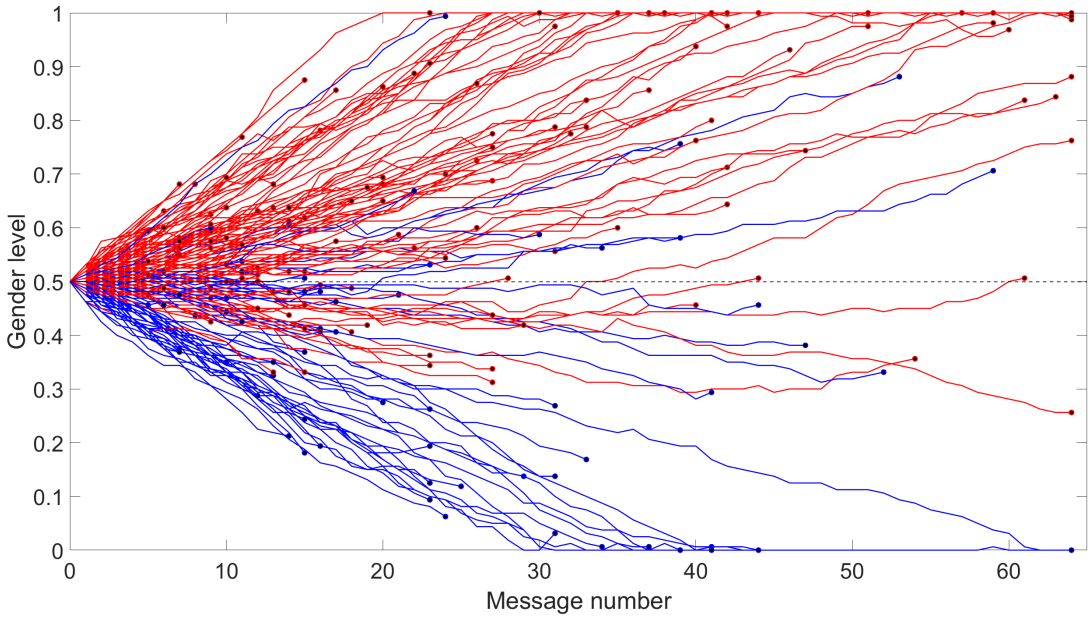

Figure 5.15: Gender level progressions using hybrid two-step gender level update mechanism with an RF classifier

Fusion	Classifier	Male accuracy			Female accuracy			Total accuracy		
		S	V	H	S	V	H	S	V	H
two-step	RF	81%	81%	76%	81%	79%	79%	81%	80%	78%

Table 5.6: End of conversation accuracies using different two-step update mechanisms

When updating the gender levels in two steps for each message, as seen in Figures 5.13 to 5.15, we observe that there not many differences compared to the one-step gender level progressions discussed in Section 5.2.4. All observations made in Section 5.2.4 also holds in this case.

The only difference is that when a two-step update mechanism is used, the gender level converges a bit slower towards 0 and 1 than when a one-step update mechanism is used. One possible explanation of this is due to the relatively low accuracy achieved when using only stylometry. As an example, consider a conversation participant whose true gender is female, and the two modalities disagree on the gender of the current message. Based on keystroke dynamics the message seems to be female with a somewhat high confidence, while based on stylometry, the message seems to be male with a slightly lower confidence. This is a typical scenario as we have observed that stylometry generally achieves both lower accuracy and lower confidence than keystroke dynamics. With a one-step update mechanism, the gender level would increase with a moderately high number because the high confidence achieved with keystroke dynamics dominates the lower confidence achieved with stylometry. With a two-step update mechanism the gender level would first increase with a somewhat high number before decreasing with a slightly lower number. The net increase would thus be lower than with a one-step update mechanism. This would naturally cause the gender level progressions using a two-step update mechanism to converge slower towards the maximum and minimum gender level.

Other than this, there are not any major differences between the number of steps used in the update mechanism. The accuracy is slightly higher than what we achieved with feature-level and score-level fusion, but the difference is too small to be considered significant. One final observation is however that when using a hybrid update mechanism (Figure 5.15), is that are no female conversation participants whose gender went all the way down to 0, and only one male conversation participant whose gender level went all the way up to 1. This shows that the false classifications, in this case, are "less false" than what we observed in Section 5.2.4, where the false classifications tended to converge rather quickly towards the wrong ends of the gender level scale.

5.2.8 Outliers

When observing the gender level progressions, we noticed a select few cases which displayed some interesting properties. We observed some instances where the gender level varied widely between both ends of the gender level scale. We call these "outliers" and are defined to be instances where the difference between the maximum and minimum gender level is more than 0.75. The gender level progressions of these instances are seen in Figure 5.16.

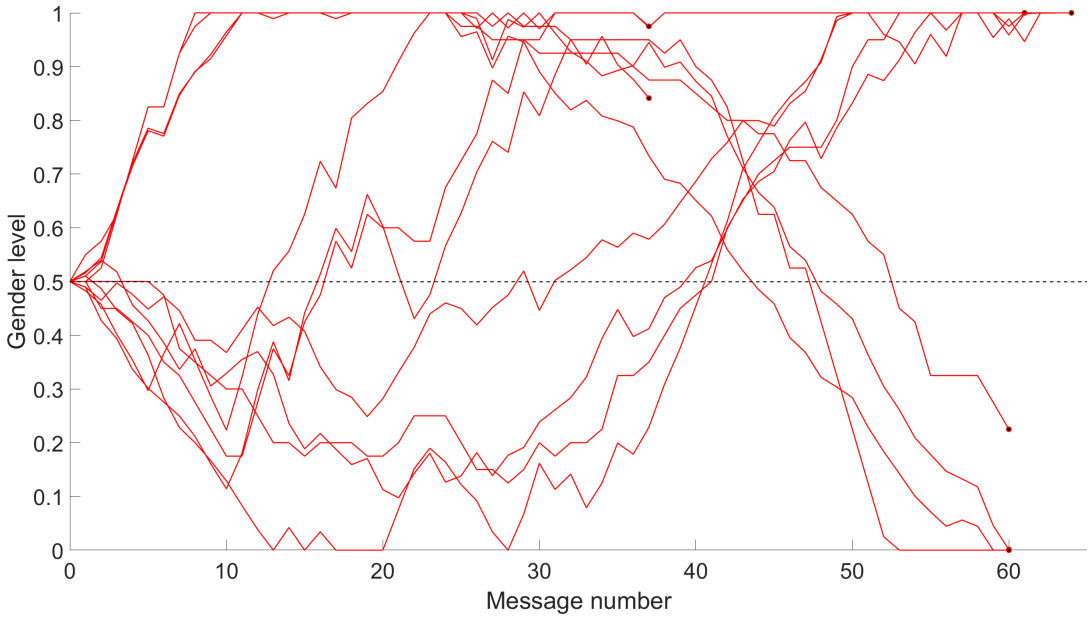

Figure 5.16: Gender level progression of conversation participants defined to be outliers

In addition to being outliers, these instances share some additional characteristics. First of all, they are all female. This could imply that females have less predictable chat behaviour than males. In addition, 1 of the outliers occurred when using the RF classifier, 2 when using the SVM classifier and 8 when using the k-NN classifier. 10 occurred when using feature-level fusion and 1 when using score-level fusion. Finally, 7 occurred when using a variable update mechanism and 4 when using a hybrid update mechanism. The clear observation is thus that most outliers occur when a female is classified using the k-NN classifier with feature-level fusion and a variable update mechanism.

To further explore the cause of the outliers, we also plotted the gender level progressions when using each modality separately. This is seen in Figures 5.17 and 5.18.

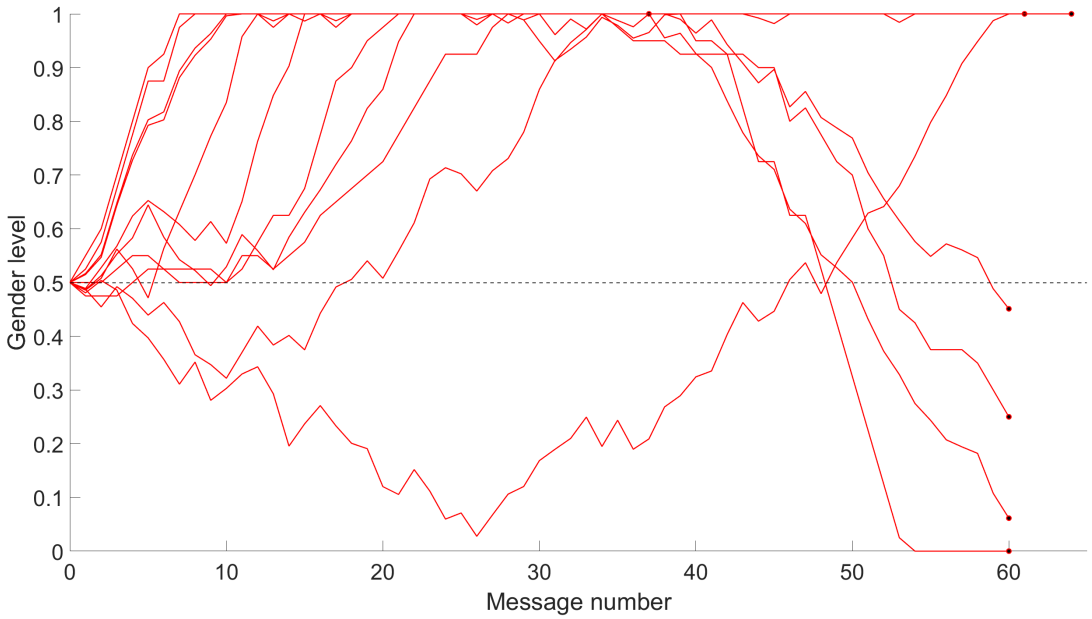

Figure 5.17: Gender level progression of conversation participants defined to be outliers using only keystroke dynamics

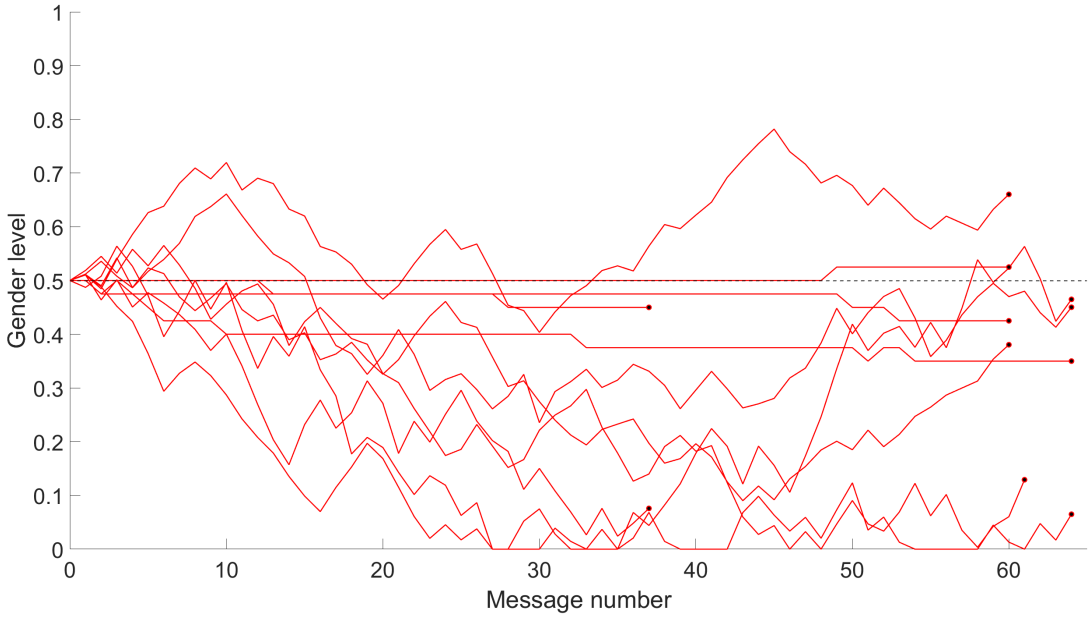

Figure 5.18: Gender level progression of conversation participants defined to be outliers using only stylometry

We can observe in Figures 5.17 and 5.18 that the outliers do not occur when using only stylometry, while some of them (4) still occur when using only keystroke dynamics. This shows that the cause of at least some of the outliers, in

our case, is that the keystroke timing information changed mid-conversation for some conversation participants. We did also observe that outliers did occur a lot more frequently when using feature-level fusion compared to score-level fusion. Due to the fact that we found keystroke dynamics to be causing some of the outliers, a possible explanation of the differences between the fusion methods is that feature-level fusion implicitly assigns heavier weights to keystroke dynamics features. As described in Section 4.1, we extracted approximately 250 keystroke dynamics features and approximately 50 stylometry features. When combining these into a single feature set, the final feature set would consist of approximately 300 features with the majority being keystroke dynamics features. In total, the collection of keystroke dynamics features would thus have a larger impact on the classification than the collection of stylometry features. This is not the case when using score-level fusion because the modalities are handled separately and only a final score is fused (see Section 4.3.2). When using score-level fusion, adjusting the weights in favour of keystroke dynamics should thus result in more outliers. To test this hypothesis, we adjusted the weights from 0.5/0.5 to 0.85/0.15 in favour of keystroke dynamics, and plotted all outliers occurring with these updated parameters. This is seen in Figure 5.19.

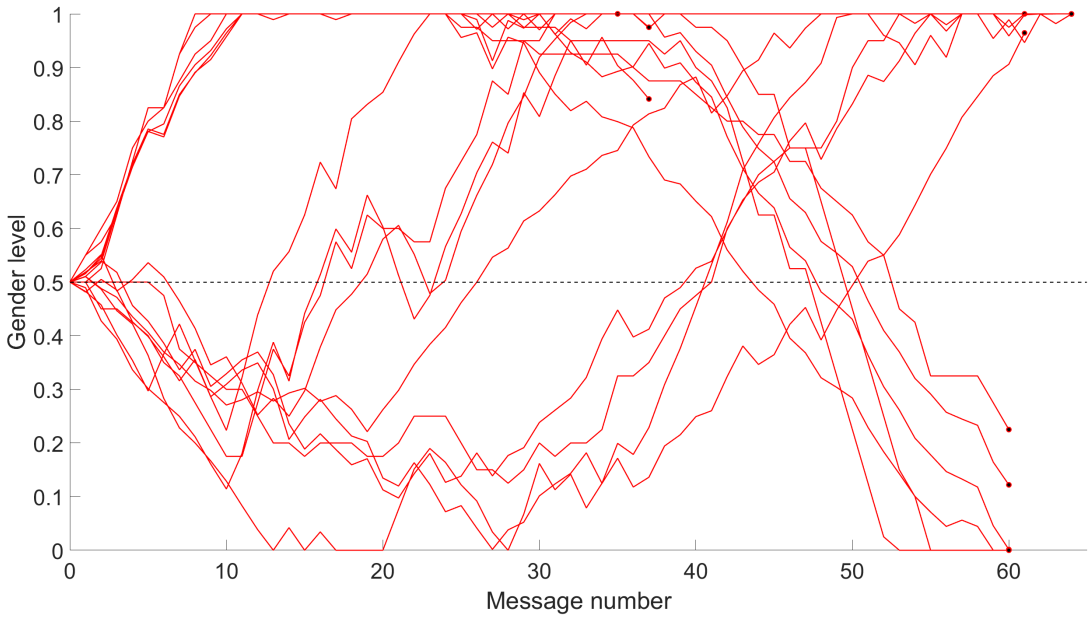

Figure 5.19: Gender level progression of conversation participants defined to be outliers with adjusted weights for score-level fusion

In Figure 5.19 there are now a total of 15 outliers. 11 of them are the same as in Figure 5.16. The additional 4 occurred when using score-level fusion due to our adjusted weights. This confirms our claim that some of the outliers are caused by feature-level fusion implicitly assigning heavier weights to keystroke dynamics features. The remaining outliers are more difficult to explain. We did not find any

specific causes of their occurrences, but as we have discussed in this section, there are several aspects that could play a role. We saw that most outliers occurred when using a k-NN classifier in combination with feature-level fusion. It is possible that there is something regarding this particular combination that potentially could lead to outliers. We also noticed that all outliers were female. This could imply that males display a more consistent chat behaviour. It is too early to draw any final conclusions, but whether there are any gender differences regarding this consistency of chat behaviour could be an interesting topic for further research.

5.2.9 Gender detection using the English dataset

As we described in Chapter 3, the data collection resulted in one Norwegian and one English dataset. We have primarily focused on the Norwegian dataset, but we also wanted to perform some analysis of the English dataset. This allows us to assess whether our gender detection scheme is sufficiently language independent to achieve strong performance in other languages than Norwegian. We used the same training parameters as with the Norwegian dataset and we calculated baseline accuracies as described in Section 5.1. These accuracies are seen in Table 5.7.

Classifier	Fusion	Accuracy
RF	Feature	55%
k-NN	Feature	59%
SVM	Feature	45%
RF	Score	55%
k-NN	Score	52%
SVM	Score	48%
RF	KD	55%
k-NN	KD	55%
SVM	KD	52%
RF	Stylometry	55%
k-NN	Stylometry	62%
SVM	Stylometry	59%

Table 5.7: Performance of classifications based on entire conversations using the English dataset

Based on Table 5.7, it is clear that the accuracy is generally much lower when using the English dataset. The accuracy is on average around 50% and when using score-level fusion, the total accuracy is actually lower than when using any of the modalities by themselves. This shows that models trained on our Norwegian dataset is not suitable for classifying the gender of chat conversation participants in our English dataset. It is however not surprising that the accuracy decreased. When changing language, it is reasonable to believe that some typing characteristics will change. One interesting observation is that the accuracy when using only stylometry is approximately the same for both the Norwegian and English

dataset. This can probably be attributed to our chosen stylometry features. Features such as emoji densities, word length and message length are less affected by changing language than features such as word selection. English and Norwegian are also both part of the same language family, Germanic languages [52], which means that they share some similarities. One could speculate that the stylometry accuracy would decrease if we tested with a language less similar to Norwegian.

The main reason the baseline accuracy is lower when using the English dataset is thus probably due to the keystroke dynamics modality. One possible explanation for this is our bigram selection. We chose to extract features from only the most common bigrams in the Norwegian dataset, but these might not necessarily be the same bigrams that are most common in the English language. Due to habit, it is also naturally to believe that common bigrams are generally typed quicker than rare bigrams. A consequence of this could be that English speakers would generally type these bigrams slower than Norwegian speakers, which would thus affect classification accuracy. Many of these issues could be resolved if we used English chat conversations for the training data, but this was not possible due to the limited size of our English dataset. Any results obtained would thus be too unreliable to be considered useful.

We also plotted the gender level progressions when using the RF classifier and a variable update mechanism, as this generally achieved the best performance when using the Norwegian dataset. These are seen in Figures 5.20 and 5.21, and Table 5.8 shows relevant accuracies associated with the gender level progressions in these figures. Even though the baseline accuracy is too low to provide any real usefulness, it can still be interesting to see how the gender levels progress.

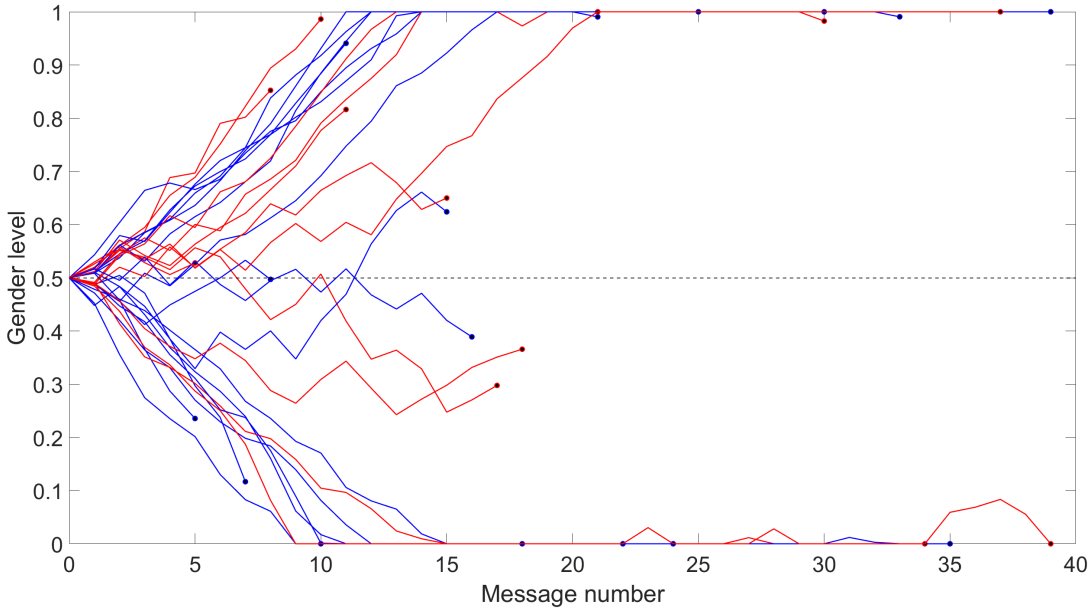

Figure 5.20: English dataset - Gender level progressions using variable gender level update mechanism and feature-level fusion with an RF classifier

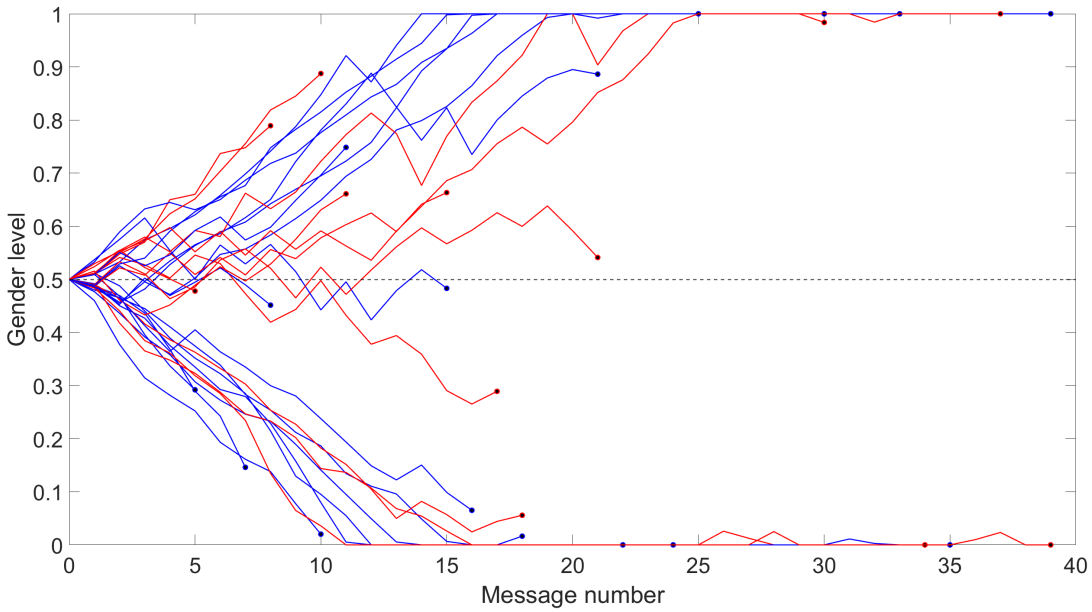

Figure 5.21: English dataset - Gender level progressions using variable gender level update mechanism and score-level fusion with an RF classifier

Fusion	Classifier	Male accuracy	Female accuracy	Total accuracy
Feature	RF	53%	67%	59%
Score	RF	59%	58%	59%

Table 5.8: End of conversation accuracies using different two-step update mechanisms

As seen in Figures 5.20 and 5.21, there is not anything particularly special with these gender level progressions. As when using the Norwegian dataset in Section 5.2.4, we can still see that the gender level progressions can be divided into the same three categories, consisting of gender level progressions that quickly converge towards the correct end of the scale, gender level progressions that quickly converge towards the wrong end of the scale and gender level progressions that tend to move in both directions. The only difference in this case, where the English dataset is used, is that there is a higher ratio of conversation participants whose gender level converges in the wrong direction. This is not surprising because of the low baseline accuracy. Other than this, they are not much different from the gender level progressions discussed in Section 5.2.4. We did not further explore the possibility of early gender detection. Because of the limitations of our English dataset and the low baseline accuracies, it would be highly unlikely to obtain any useful results.

Chapter 6

Conclusion and future research

6.1 Conclusion

In this project, we have explored the possibility of performing gender detection early in a conversation and we have found clear indications that this should indeed be possible.

When using our gender level system, we found that the vast majority of conversation participants in our test dataset depicted a highly gender typical chat behaviour in regard to the extracted keystroke dynamics and stylometry features. This means that most females wrote predominantly messages considered to be female and that most males wrote predominantly messages considered to be male. A consequence of this was that we were able to perform gender detection early without much accuracy loss. Our results showed that it was possible to halve the average number of messages the classification is based upon (from 28 to 14) without any accuracy loss. By further reducing the average number of messages to 5, the accuracy loss was still <5 percentage points in most cases. Any further reduction of the average number of messages generally increased the accuracy loss too much to be considered tolerable.

These results also showed that our gender level system performed well in regards to determining when to perform the classification. As the results above describe, the gender level system allowed us to maintain both performance measures in most cases. We tested both absolute and stability thresholds and found no significant differences between them.

We performed the analysis using two methods of fusion (feature-level and score-level). We did not find any significant differences between them, which could indicate that both feature-level and score-level fusion is appropriate for early gender detection. One interesting observation is however that feature-level fusion generated more outliers. We also observed that keystroke dynamics generally performed better than stylometry for the purpose of gender detection. Finally, we tested our gender detection scheme with an English dataset, which resulted in significantly lower accuracy. Although the English dataset was small, it is still a sign that our gender detection scheme is not language independent.

6.2 Future research

In this research project, we have found promising results regarding the possibility of early gender detection. The topic is however by no means fully explored, and there are still several aspects that could be suspect for future research. This chapter will describe some of them.

One of the limitations in this project is that we have only used a dataset consisting of adult subjects. Research has already shown that there exist differences between age groups in regards to keystroke dynamics and stylometry [15, 40]. This could potentially also affect the capabilities of early gender detection. An interesting topic could thus be to explore whether children and adults share similar properties when it comes to early gender detection.

Another limitation was that the small size of the English dataset, made it difficult to obtain any reliable results when using that dataset. Further testing with a larger English dataset, or any other language of interest, could thus be of interest for future research.

One of our findings in Chapter 5 was that all of the most extreme outliers were female. This could imply that males are more consistent in typing behaviour. Due to the limited amount of data, it was however not possible to draw a final conclusion. It could be interesting to research whether our findings are correct or simply coincidental.

Finally, due to the finite scope of this project, it was not practically possible to perform testing with all possible combinations of parameters used during analysis. Further research results could possibly be obtained by adjusting these, such as trying other machine learning models, selecting different features or making changes to the gender level system. Especially further research regarding the potential of stability thresholds could be interesting. We did not find any significant advantages of using stability thresholds, but due to our relatively small dataset, more research should be performed to either confirm or disprove these results. An additional example could be to base the final gender classification on several sub-classifications performed during the course of a conversation. We saw in Section 6.1 that we can achieve good accuracy when basing the classification on 5 messages. One could than imagine a system where the gender level is updated as usual for the first 5 messages, before a gender classification is performed and the gender level is reset back to its default value of 0.5. This process is then repeated for the next 5 messages and so on. A final classification can then be based on a majority voting of all the performed classifications. It is possible that this could affect accuracy and also prevent the occurrence of some outliers as sudden large changes to gender level would be nullified when resetting the gender level. Both these aspects make it a fitting and interesting topic for future research.

Bibliography

[1] G. O. Jibril, K. S. Mikalsen and M. Haugli, *Dom i norgeshistoriens største overgrepssak: 16 år i fengsel*, Accessed at 03.11.2020, 2019. [Online]. Available: https://www.nrk.no/osloogviken/dom-i-norgeshistoriens-storste-overgrepssak_-16-ar-i-fengsel-1.14607297.

[2] M. Ogre, *-Gjør som jeg sier, ellers ødelegger jeg livet ditt*, Accessed at 03.11.2020, 2018. [Online]. Available: https://www.nrk.no/norge/ung-mann-tok-livet-sitt-etter-a-ha-blitt-utsatt-for-seksuell-utpressing-1.13943519.

[3] ABC News, *Parents: Cyber bullying led to teen's suicide*, Accessed at 13.11.2020, 2009. [Online]. Available: https://abcnews.go.com/GMA/story?id=3882520&page=1.

[4] V. Borgersen, *Minst 150 nordmenn robbes for 200 mill. i året av kjærlighetssvindlere på nettet*, Accessed at 13.11.2020, 2018. [Online]. Available: https://www.aftenposten.no/norge/i/G1Q5Wq/minst-150-nordmenn-robbes-for-200-mill-i-aaret-av-kjaerlighetssvindlere.

[5] Z. Rui and Z. Yan, 'A survey on biometric authentication: Toward secure and privacy-preserving identification,' *IEEE Access*, vol. 7, pp. 5994–6009, 2018.

[6] P. S. Prasad, B. S. Devi, M. J. Reddy and V. K. Gunjan, 'A survey of fingerprint recognition systems and their applications,' in *International Conference on Communications and Cyber Physical Engineering 2018*, Springer, 2018, pp. 513–520.

[7] Z. Mahmood, N. Muhammad, N. Bibi and T. Ali, 'A review on state-of-the-art face recognition approaches,' *Fractals*, vol. 25, no. 02, p. 1 750 025, 2017.

[8] H. N. M. Shah, M. Z. Ab Rashid, M. F. Abdollah, M. N. Kamarudin, C. K. Lin and Z. Kamis, 'Biometric voice recognition in security system,' *Indian journal of Science and Technology*, vol. 7, no. 2, p. 104, 2014.

[9] M. Faundez-Zanuy, 'Signature recognition state-of-the-art,' *IEEE aerospace and electronic systems magazine*, vol. 20, no. 7, pp. 28–32, 2005.

[10] P. S. Teh, A. B. J. Teoh and S. Yue, 'A survey of keystroke dynamics biometrics,' *The Scientific World Journal*, vol. 2013, 2013.

[11] Y. Zhong and Y. Deng, 'A survey on keystroke dynamics biometrics: Approaches, advances, and evaluations,' *Recent Advances in User Authentication Using Keystroke Dynamics Biometrics*, pp. 1–22, 2015.

[12] A. Dantcheva, P. Elia and A. Ross, 'What else does your biometric data reveal? a survey on soft biometrics,' *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 3, pp. 441–467, 2015.

[13] A. A. Buker, G. Roffo and A. Vinciarelli, 'Type like a man! inferring gender from keystroke dynamics in live-chats,' *IEEE Intelligent Systems*, vol. 34, no. 6, pp. 53–59, 2019.

[14] I. Tsimperidis, A. Arampatzis and A. Karakos, 'Keystroke dynamics features for gender recognition,' *Digital Investigation*, vol. 24, pp. 4–10, 2018.

[15] A. Pentel, 'Predicting age and gender by keystroke dynamics and mouse patterns,' in *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*, 2017, pp. 381–385.

[16] G. Li, P. R. Borj, L. Bergeron and P. Bours, 'Exploring keystroke dynamics and stylometry features for gender prediction on chat data,' in *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, IEEE, 2019, pp. 1049–1054.

[17] S. Z. S. Idrus, E. Cherrier, C. Rosenberger and P. Bours, 'Soft biometrics for keystroke dynamics: Profiling individuals while typing passwords,' *Computers & Security*, vol. 45, pp. 147–155, 2014.

[18] S. Roy, U. Roy and D. Sinha, 'Identifying soft biometric traits through typing pattern on touchscreen phone,' in *Annual Convention of the Computer Society of India*, Springer, 2018, pp. 546–561.

[19] M. Antal and G. Nemes, 'Gender recognition from mobile biometric data,' in *2016 IEEE 11th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, IEEE, 2016, pp. 243–248.

[20] B. Plank, 'Predicting authorship and author traits from keystroke dynamics,' in *Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media*, 2018, pp. 98–104.

[21] D. G. Brizan, A. Goodkind, P. Koch, K. Balagani, V. V. Phoha and A. Rosenberg, 'Utilizing linguistically enhanced keystroke dynamics to predict typist cognition and demographics,' *International Journal of Human-Computer Studies*, vol. 82, pp. 57–68, 2015.

[22] G. Biau and E. Scornet, 'A random forest guided tour,' *Test*, vol. 25, no. 2, pp. 197–227, 2016.

[23] W. S. Noble, 'What is a support vector machine?' *Nature biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2006.

[24] I. Rish *et al.*, 'An empirical study of the naive bayes classifier,' in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, 2001, pp. 41–46.

[25] F. Murtagh, 'Multilayer perceptrons for classification and regression,' *Neurocomputing*, vol. 2, no. 5-6, pp. 183–197, 1991.

[26] M. J. Orr, 'Introduction to radial basis function networks,' *Technical Report, center for cognitive science, University of Edinburgh*, 1996.

[27] D. G. Kleinbaum, K. Dietz, M. Gail, M. Klein and M. Klein, *Logistic regression*. Springer, 2002.

[28] A. Mucherino, P. J. Papajorgji and P. M. Pardalos, 'K-nearest neighbor classification,' in *Data mining in agriculture*, Springer, 2009, pp. 83–106.

[29] S. Ruggieri, 'Efficient c4. 5 [classification algorithm],' *IEEE transactions on knowledge and data engineering*, vol. 14, no. 2, pp. 438–444, 2002.

[30] J. B. Lang, 'On the comparison of multinomial and poisson log-linear models,' *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 253–266, 1996.

[31] J. Friedman, T. Hastie, R. Tibshirani *et al.*, 'Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors),' *Annals of statistics*, vol. 28, no. 2, pp. 337–407, 2000.

[32] K. Lagutina, N. Lagutina, E. Boychuk, I. Vorontsova, E. Shliakhtina, O. Belyaeva, I. Paramonov and P. Demidov, 'A survey on stylometric text features,' in *2019 25th Conference of Open Innovations Association (FRUCT)*, IEEE, 2019, pp. 184–195.

[33] W. Daelemans, 'Explanation in computational stylometry,' in *International conference on intelligent text processing and computational linguistics*, Springer, 2013, pp. 451–462.

[34] T. Kucukyilmaz, B. B. Cambazoglu, C. Aykanat and F. Can, 'Chat mining for gender prediction,' in *International conference on advances in information systems*, Springer, 2006, pp. 274–283.

[35] S. Ashraf, O. Javed, M. Adeel, H. Iqbal and R. M. A. Nawab, 'Bots and gender prediction using language independent stylometry-based approach.,' in *CLEF (Working Notes)*, 2019.

[36] C. S. Montero, M. Munezero and T. Kakkonen, 'Investigating the role of emotion-based features in author gender classification of text,' in *International Conference on Intelligent Text Processing and Computational Linguistics*, Springer, 2014, pp. 98–114.

[37] F. Huang, C. Li and L. Lin, 'Identifying gender of microblog users based on message mining,' in *International conference on web-age information management*, Springer, 2014, pp. 488–493.

[38] M. Fatima, K. Hasan, S. Anwar and R. M. A. Nawab, 'Multilingual author profiling on facebook,' *Information Processing & Management*, vol. 53, no. 4, pp. 886–904, 2017.

[39] N. Cheng, R. Chandramouli and K. Subbalakshmi, 'Author gender identification from text,' *Digital Investigation*, vol. 8, no. 1, pp. 78–88, 2011.

[40] K. Surendran, O. Harilal, P. Hrudya, P. Poornachandran and N. Suchetha, 'Stylometry detection using deep learning,' in *Computational Intelligence in Data Mining*, Springer, 2017, pp. 749–757.

[41] S. Albawi, T. A. Mohammed and S. Al-Zawi, 'Understanding of a convolutional neural network,' in *2017 International Conference on Engineering and Technology (ICET)*, Ieee, 2017, pp. 1–6.

[42] N. Bhargava, G. Sharma, R. Bhargava and M. Mathuria, 'Decision tree analysis on j48 algorithm for data mining,' *Proceedings of international journal of advanced research in computer science and software engineering*, vol. 3, no. 6, 2013.

[43] A. J. Myles, R. N. Feudale, Y. Liu, N. A. Woody and S. D. Brown, 'An introduction to decision tree modeling,' *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 18, no. 6, pp. 275–285, 2004.

[44] L. M. Dinca and G. P. Hancke, 'The fall of one, the rise of many: A survey on multi-biometric fusion methods,' *IEEE Access*, vol. 5, pp. 6247–6289, 2017.

[45] M. Ghayoumi, 'A review of multimodal biometric systems: Fusion methods and their applications,' in *2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS)*, IEEE, 2015, pp. 131–136.

[46] P. Bours, 'Continuous keystroke dynamics: A different perspective towards biometric evaluation,' *Information Security Technical Report*, vol. 17, no. 1-2, pp. 36–43, 2012.

[47] P. Bours, S. Mondal, Y. Zhong and Y. Deng, 'Continuous authentication with keystroke dynamics,' *Norwegian Information Security Laboratory NISlab*, pp. 41–58, 2015.

[48] MATLAB, *Version 9.10.0.1602886 (R2021a)*, Natick, Massachusetts: The MathWorks Inc., 2021.

[49] G. Chandrashekar and F. Sahin, 'A survey on feature selection methods,' *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.

[50] T. Neal, K. Sundararajan, A. Fatima, Y. Yan, Y. Xiang and D. Woodard, 'Surveying stylometry techniques and applications,' *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, pp. 1–36, 2017.

[51] L. Friedman and O. V. Komogortsev, 'Assessment of the effectiveness of seven biometric feature normalization techniques,' *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 10, pp. 2528–2536, 2019.

[52] J. O. Askedal, *Germanske språk*, Accessed at 18.05.2020, 2020. [Online]. Available: `https://snl.no/germanske_spr%5C%C3%5C%A5k`.

Appendix A

Selected bigrams

The nabla symobol ∇ represents the space character

Bigram	r∇	er	e∇	t∇	en	∇s	de	∇d	n∇	g∇	et
Relative frequency	3.0%	2.7%	2.4%	2.3%	2.1%	2.0%	1.9%	1.7%	1.6%	1.6%	1.5%

Table A.1: Bigram 1-11 and their relative frequency

Bigram	å∇	∇m	∇e	∇h	me	te	∇f	eg	,∇	je	re
Relative frequency	1.3%	1.3%	1.3%	1.3%	1.1%	1.1%	1.0%	1.0%	1.0%	0.9%	0.9%

Table A.2: Bigram 12-22 and their relative frequency

Bigram	ha	an	∇i	ar	li	ke	el	∇v	∇j	∇o	ne
Relative frequency	0.9%	0.8%	0.8%	0.8%	0.8%	0.8%	0.8%	0.8%	0.8%	0.8%	0.8%

Table A.3: Bigram 23-33 and their relative frequency

Bigram	or	∇p	∇t	le	tt	i∇	∇b	st	in	∇a	ik
Relative frequency	0.7%	0.7%	0.7%	0.7%	0.7%	0.7%	0.7%	0.6%	0.6%	0.6%	0.6%

Table A.4: Bigram 34-44 and their relative frequency

Bigram	sk	ve	∇k	se	ti	kk	he
Relative frequency	0.6%	0.6%	0.6%	0.6%	0.6%	0.6%	0.5%

Table A.5: Bigram 45-51 and their relative frequency

Appendix B

Gender level progressions

This appendix includes all gender level progressions used or mentioned in Section 5.2.4, in addition to a table summarizing associated accuracies.

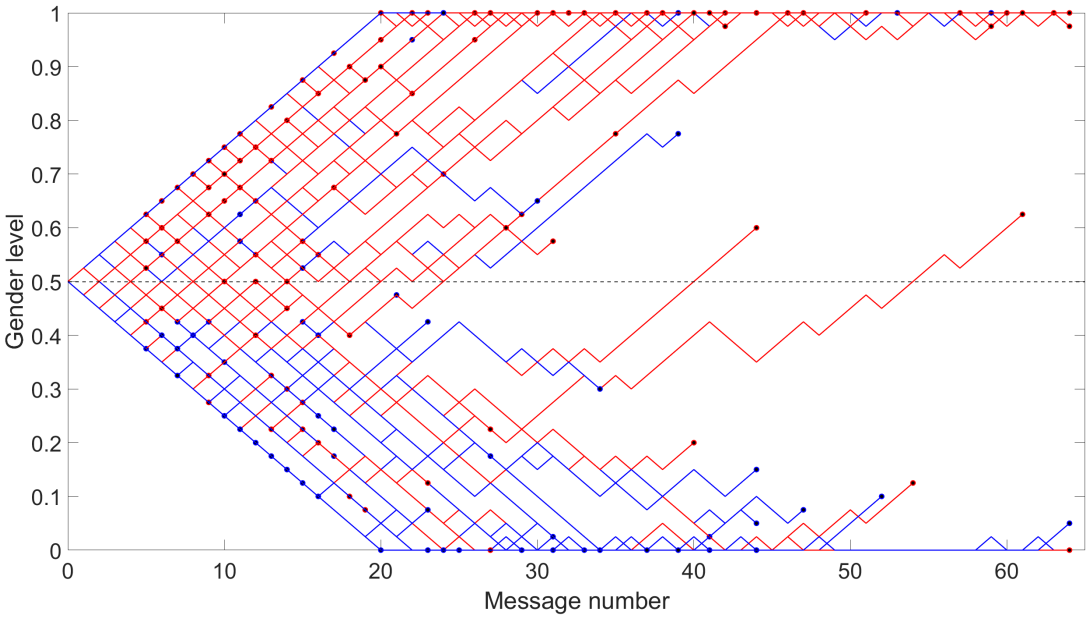

Figure B.1: Gender level progressions using static gender level update mechanism and score-level fusion with the RF classifier

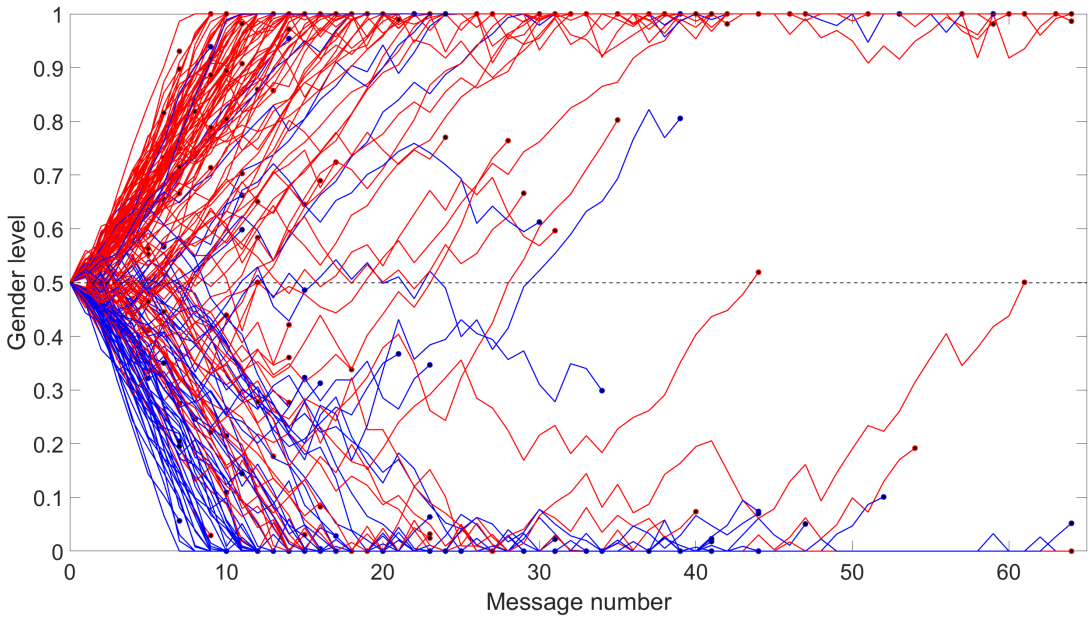

Figure B.2: Gender level progressions using variable gender level update mechanism and score-level fusion with the RF classifier

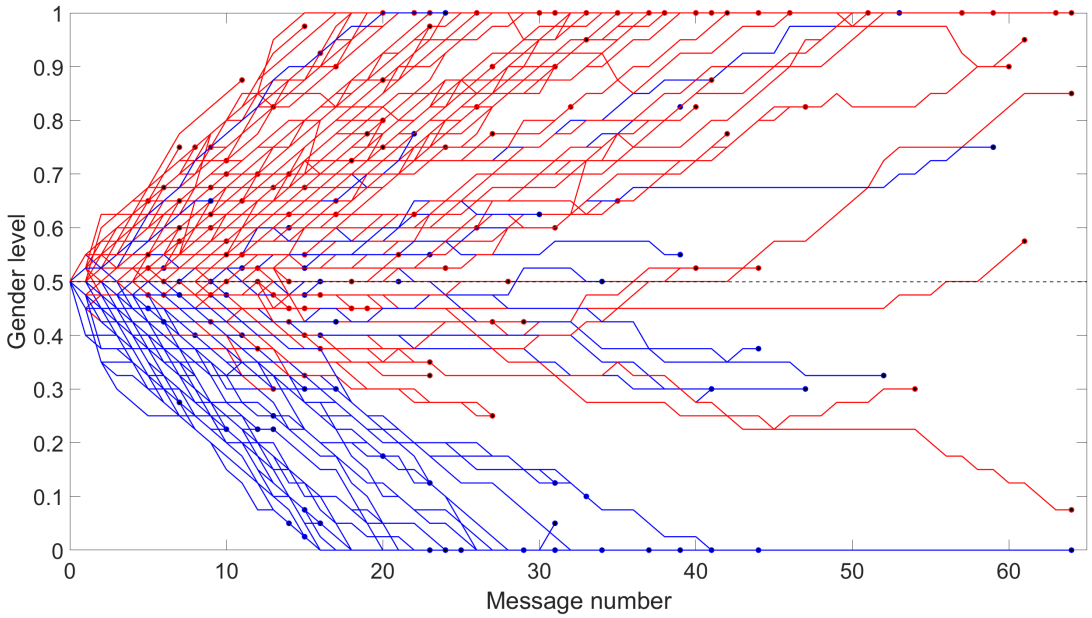

Figure B.3: Gender level progressions using hybrid gender level update mechanism and score-level fusion with the RF classifier

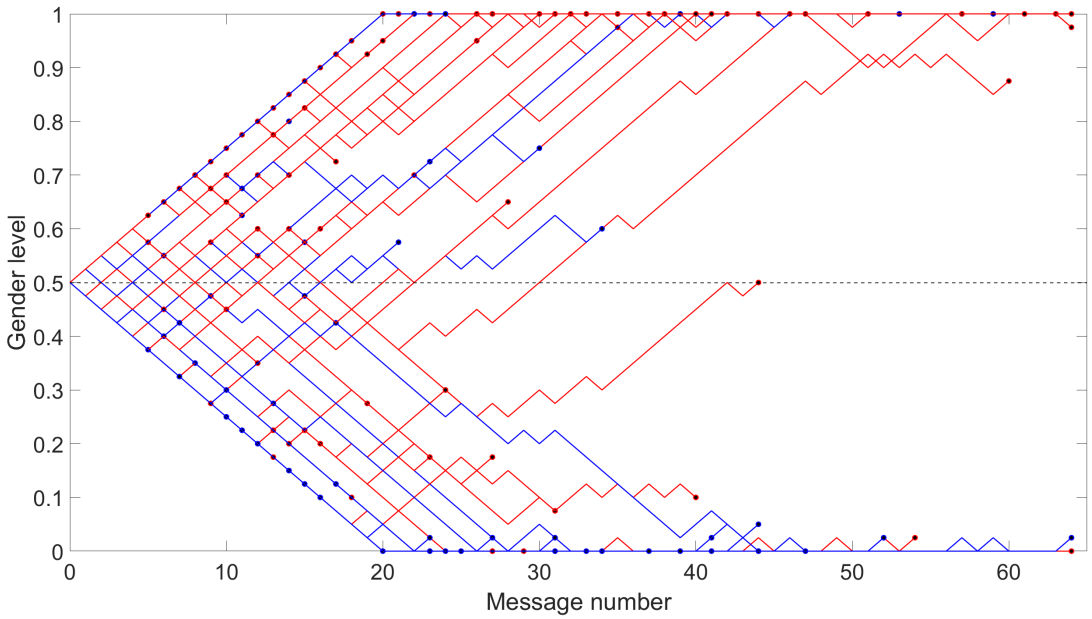

Figure B.4: Gender level progressions using static gender level update mechanism and feature-level fusion with the RF classifier

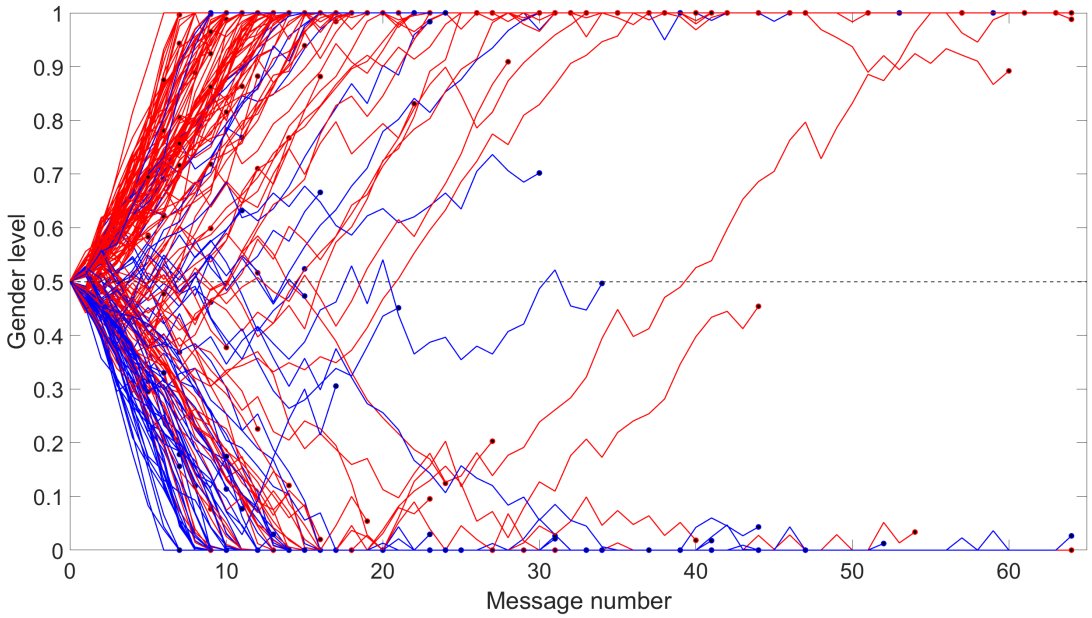

Figure B.5: Gender level progressions using variable gender level update mechanism and feature-level fusion with the RF classifier

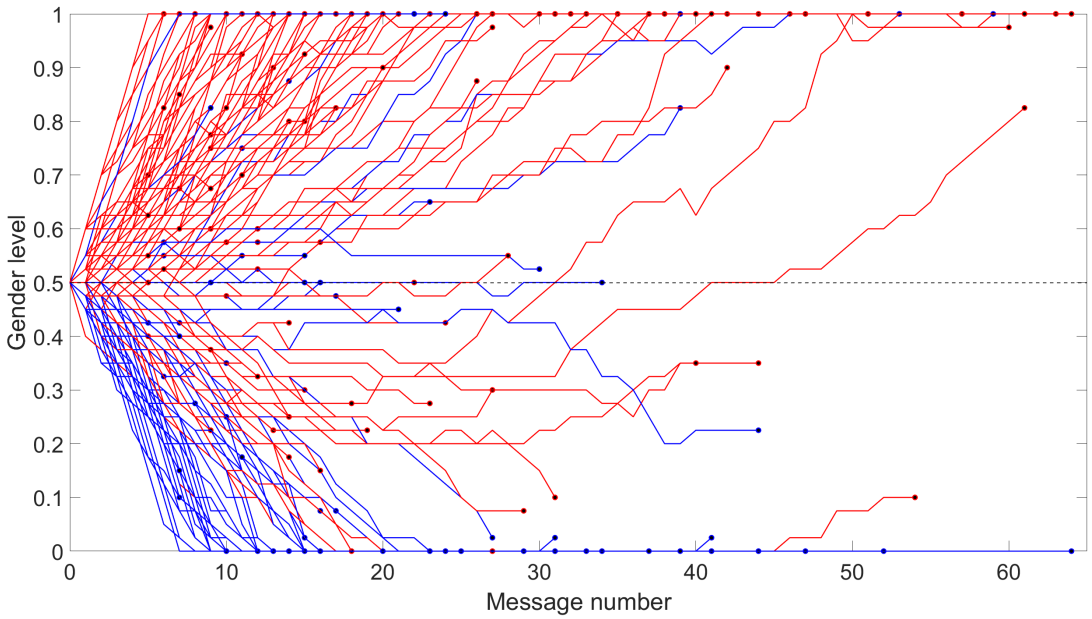

Figure B.6: Gender level progressions using hybrid gender level update mechanism and feature-level fusion with the RF classifier

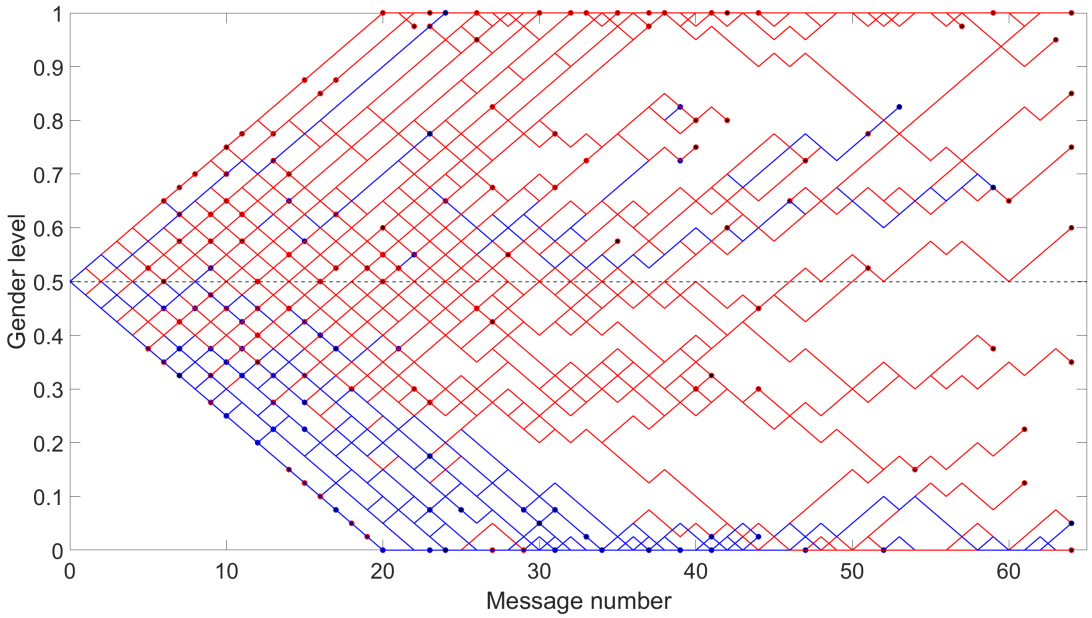

Figure B.7: Gender level progressions using static gender level update mechanism and score-level fusion with the k-NN classifier

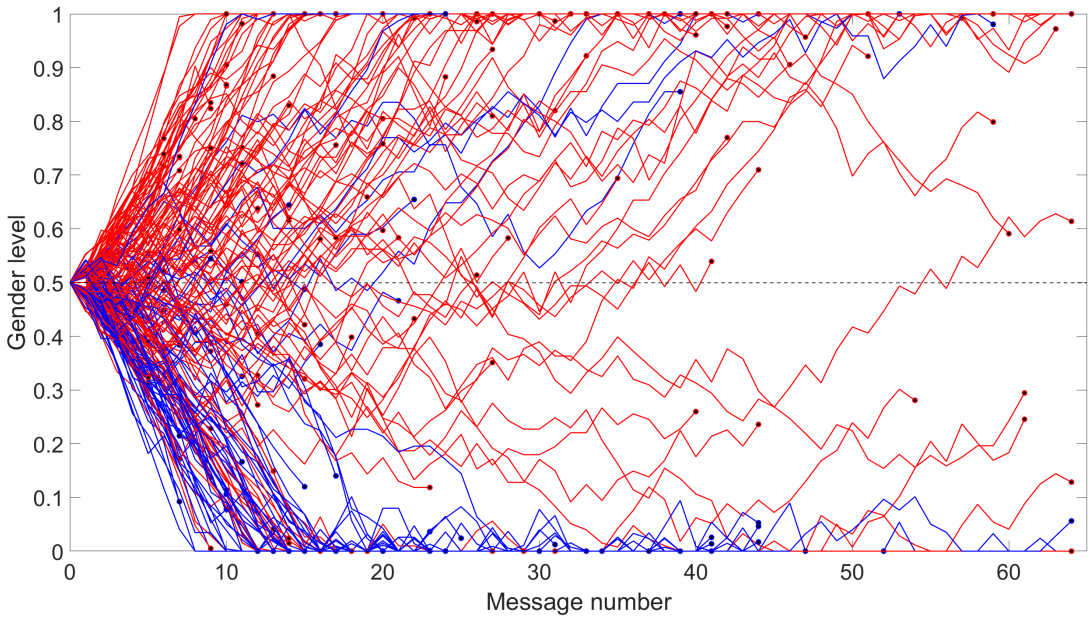

Figure B.8: Gender level progressions using variable gender level update mechanism and score-level fusion with the k-NN classifier

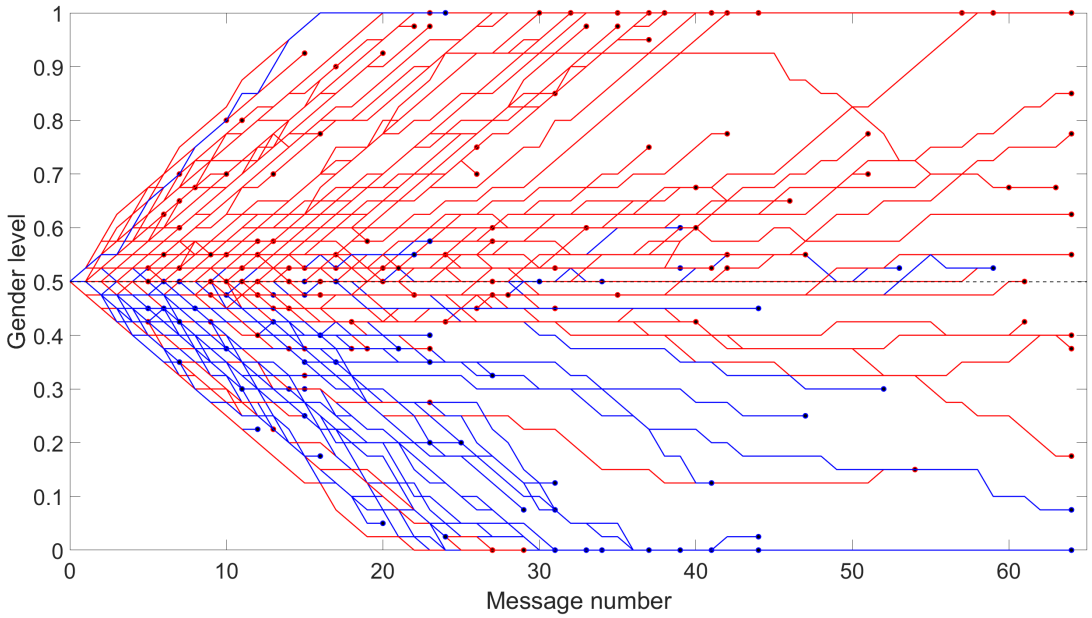

Figure B.9: Gender level progressions using hybrid gender level update mechanism and score-level fusion with the k-NN classifier

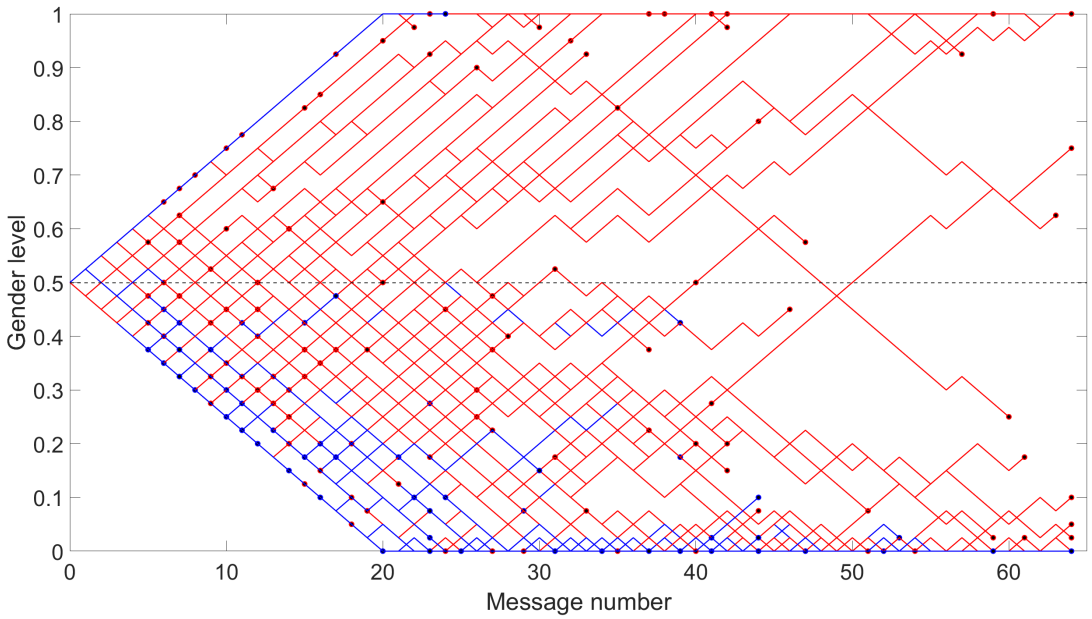

Figure B.10: Gender level progressions using static gender level update mechanism and feature-level fusion with the k-NN classifier

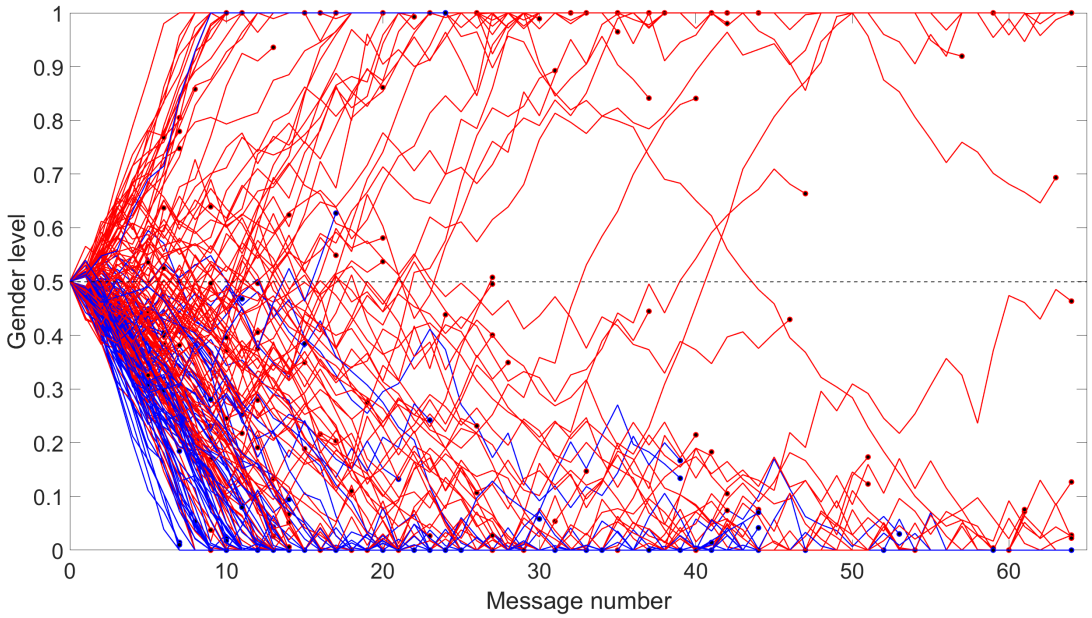

Figure B.11: Gender level progressions using variable gender level update mechanism and feature-level fusion with the k-NN classifier

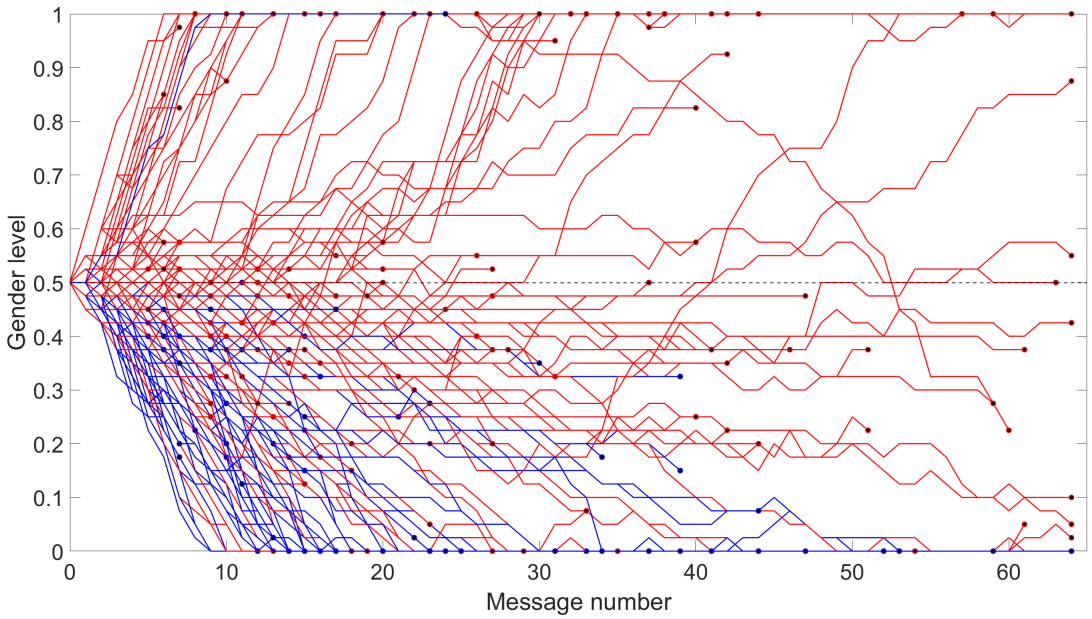

Figure B.12: Gender level progressions using hybrid gender level update mechanism and feature-level fusion with the k-NN classifier

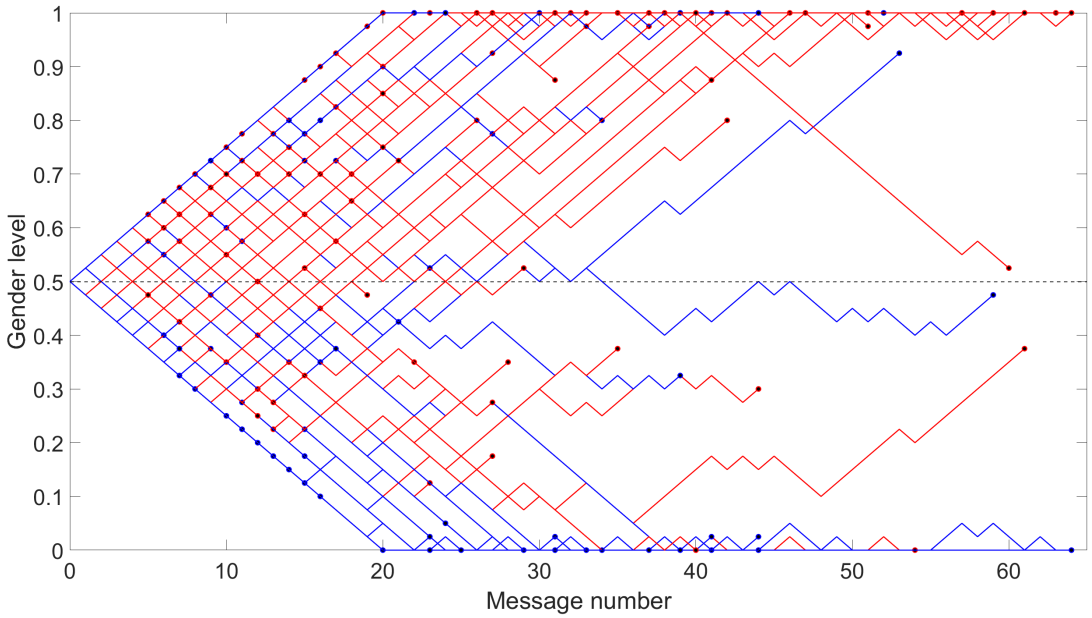

Figure B.13: Gender level progressions using static gender level update mechanism and score-level fusion with the SVM classifier

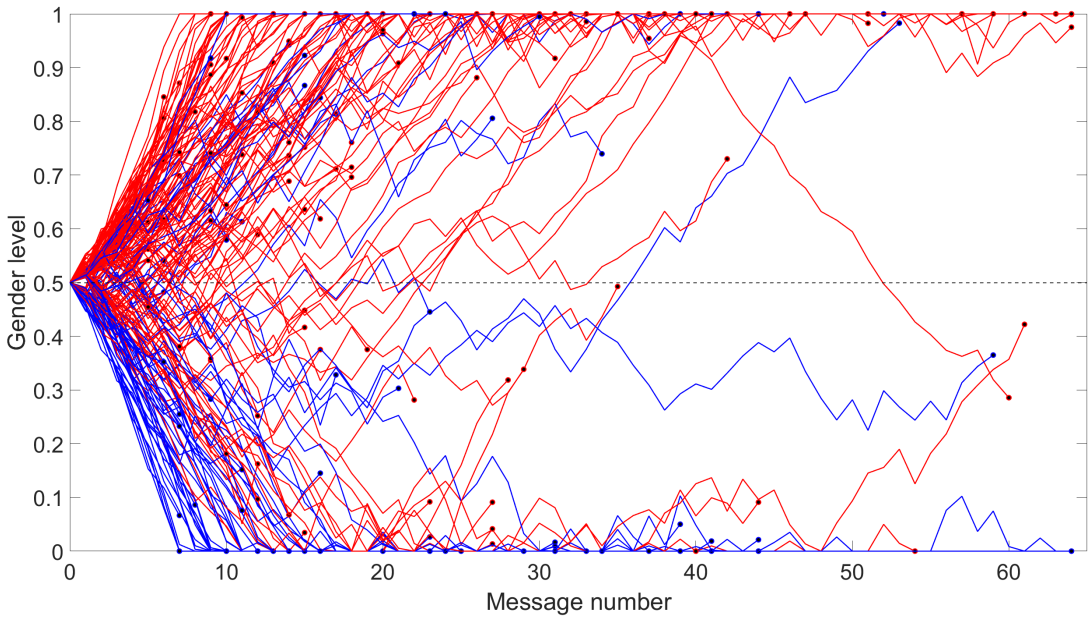

Figure B.14: Gender level progressions using variable gender level update mechanism and score-level fusion with the SVM classifier

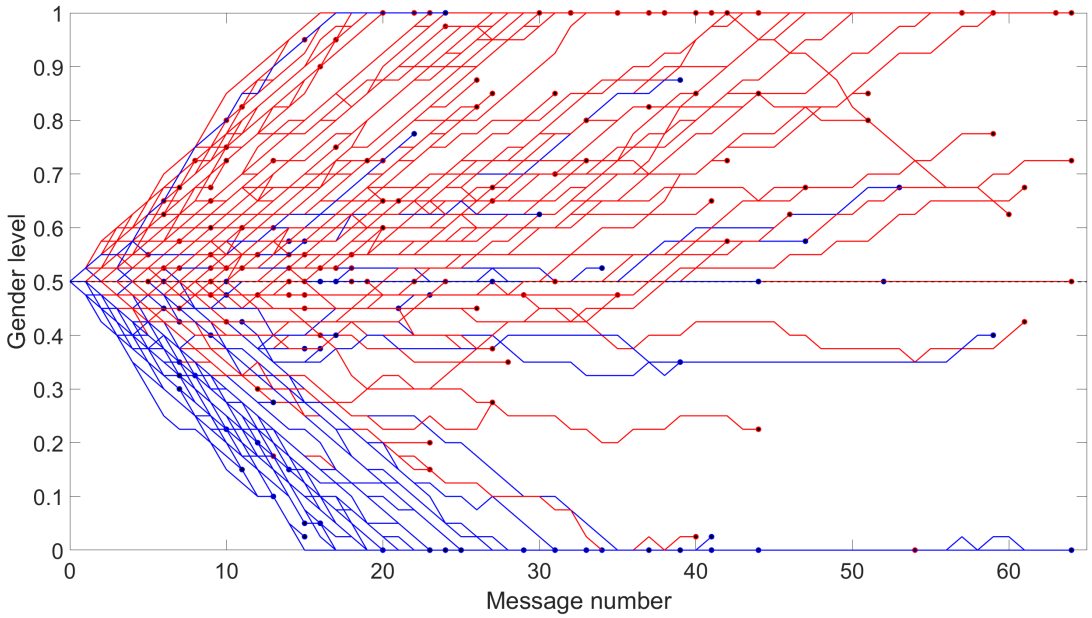

Figure B.15: Gender level progressions using hybrid gender level update mechanism and score-level fusion with the SVM classifier

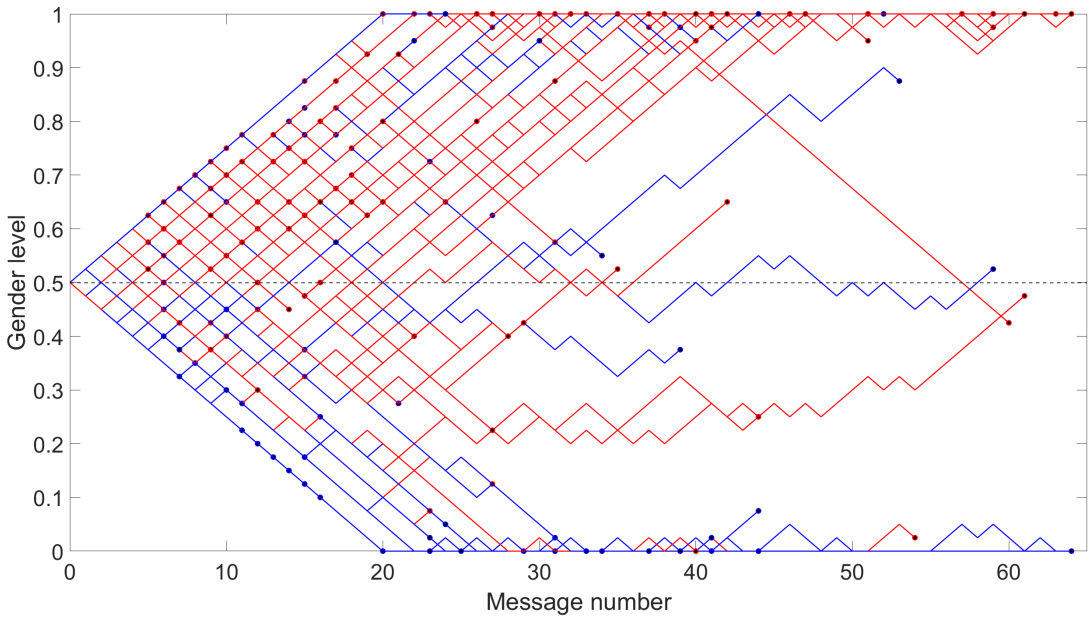

Figure B.16: Gender level progressions using static gender level update mechanism and feature-level fusion with the SVM classifier

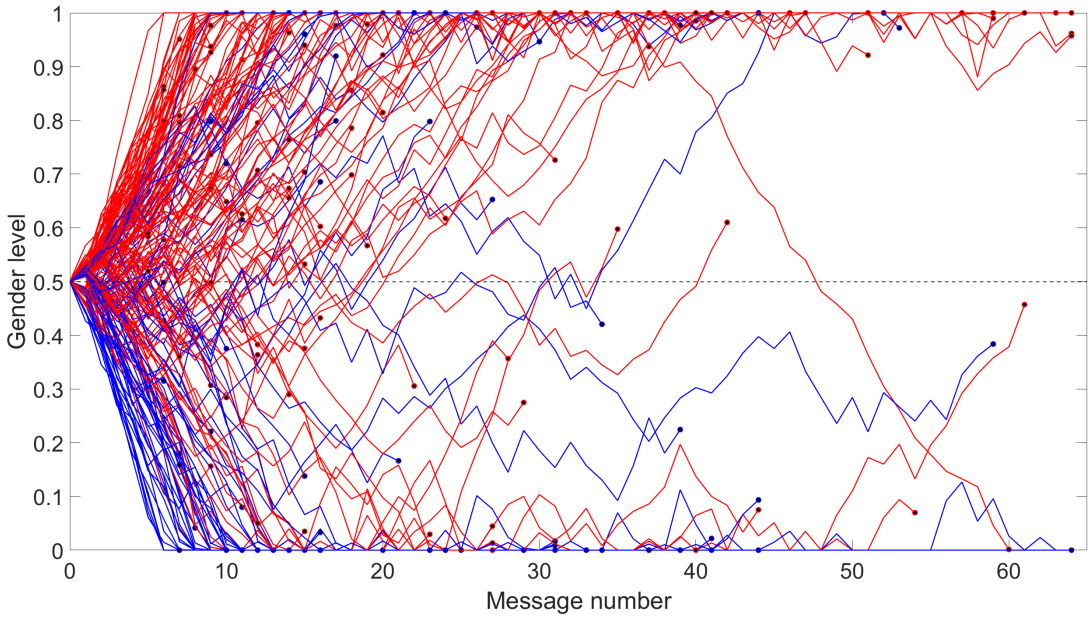

Figure B.17: Gender level progressions using variable gender level update mechanism and feature-level fusion with the SVM classifier

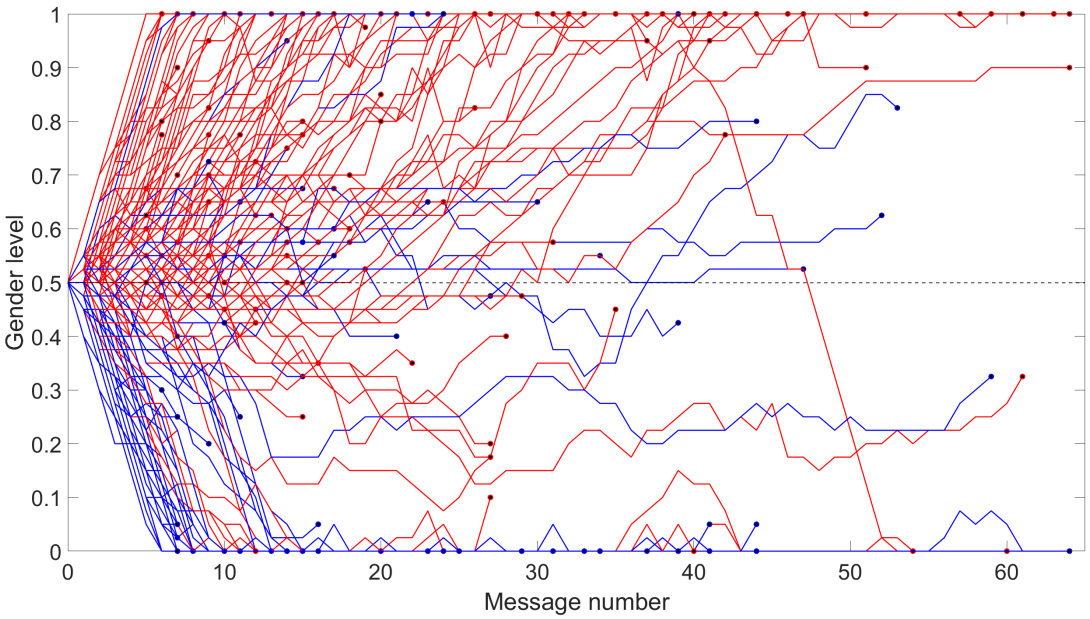

Figure B.18: Gender level progressions using hybrid gender level update mechanism and feature-level fusion with the SVM classifier

Fusion	Classifier	Male accuracy			Female accuracy			Total accuracy		
		S	V	H	S	V	H	S	V	H
Feature	RF	74%	76%	72%	79%	79%	77%	77%	78%	75%
Score	RF	79%	81%	69%	80%	80%	78%	80%	80%	75%
Feature	k-NN	99%	97%	97%	30%	37%	37%	53%	57%	57%
Score	k-NN	82%	84%	71%	60%	68%	60%	67%	73%	64%
Feature	SVM	66%	71%	69%	81%	80%	78%	76%	77%	75%
Score	SVM	66%	69%	69%	80%	78%	73%	75%	75%	71%

Table B.1: End of conversation accuracies using different update mechanisms and methods of fusion

NTNU

Norwegian University of
Science and Technology