university of groningen

faculty of mathematics and natural sciences

Authorship identification for online texts

Master thesis

First supervisor Prof. Marco Aiello

Second supervisor Prof. Michael Biehl

Company Kalooga

Tuan Luu Dinh

My special thanks go to
Prof. Marco Aiello, who facilitated me for completing this thesis,
Mr. Mathijs Homminga, CTO of Kalooga, who provided me this opportunity,
and to Prof. Michael B., Steven M.V. and Tim F. for their support.

Abstract
With the rapid development of the Internet and the proliferation of the number of its users, the Internet is becoming an ideal environment for the criminal activities such as online scams and child sexual abuse [1.10]. Usually, it is difficult to trace the culprits since they will not use their real identity on the Internet, yet by examining the evidence left by the culprits such as scam emails or abusive messages, with the support of the authorship analysis research, it can help to reveal more information about the original writers of those messages by analyzing the writing style, increasing the probability of finding the criminals. This thesis is divided into two parts. In the first part, we will search for the methods and techniques that are being used in the authorship identification research area and then implementing them into a workable prototype. Then the implemented prototype will be examined and evaluated for the qualitative (the accuracy level in identifying the correct original writers of the given texts), and quantitative (the prototype's execution time) attributes. In the later part of this thesis, four different improvements for the existing methods and techniques will be introduced and tested for their performance. The performed experiments show that in the authorship identification task, the support vector machine classifier can archive a better result from 10% to 30% compares to the other types of classifier in the qualitative experiment. Furthermore, the authorship identification task is not just a supervised learning problem; the results from four different improvements showed that the unsupervised learning technique can also help to improve the performance of the existing methods and techniques. In general, the accuracy level of the prototype, after applying the proposed improvements is improved by 05% to 10% while the performance is improved by 20% to 50%.

Abbreviation	Definition
SVM	Support vector machine
NN	Neural network
SOM	Self-organizing map
BG	Brennan-Greenstadt Adversarial
RBF	Radial basic function

Table of contents

I. Introduction

1.1 Context

Authorship analysis is a research field which focuses on the relationship between writers and their writings. Researchers in this field believe that the writing styles are different from person to person since different persons are different in their word choices, sentence structures and usage of punctuations [1.2].

With the significant increment in the amount of Internet users, combining to the openness attribute of the Internet itself; as a result, the Internet users can be, or pretend to be anyone in this virtual world. One of the famous examples is the Bit-coin [5.4], an online virtual cash trading system. Many people use Bit-coin, but none of them truly knows who is the creator of it, thanks to the anonymity allowed on the Internet. Since none of individual or government can control the Internet, it becomes an ideal place for criminals. In recent years, the Internet has been a place of the fraudulent schemes, such as email scams or online anonymity abuses [1.10]. A common point in the fraudulent schemes such as email scams and online anonymity abuses is that the culprits usually leave behind some evidences, such as emails and forum posts, which can become potential textual traces of their identities.

Early researchers in the authorship analysis from linguistics discovered that most of people have their unique stylistic discriminators and characteristics, just like their biological fingerprints. According to Holmes D [1.1], those unique idiosyncrasies are different from person to person but will remain approximately invariant among the writings of the same individual as their unique writing styles. With this concept, different analysis features and techniques are being researched and developed by linguistic, scientist and have gained notable results in the authorship identification research field. Not far from now, when the features and techniques are perfected, we will be able to find the right answer of the question about who wrote the Bit-coin white-paper.

With the progression in the authorship analysis research field, different features and techniques have been developed in order to support for the research. At Kalooga, a Dutch company who is specializing in providing online visual relevance information and is the main supporter for this thesis, Kalooga has been aware of the development in the authorship analysis research field, thus they want to focus more on analyzing online texts, which can be a crawled blog article, a random message on a discussion forum, or an email that is revealed on the Internet. Kalooga is interested in identifying the original writer of a piece of text left by an anonymous individual by comparing it to the texts that were written by already known people. Besides the Kalooga company with their own interests in the authorship analysis research, this research field also draws the attention of other companies and organizations in different areas, such as in organizations that do cyber criminals investigation who want to reveal the identity of anonymous people on the Internet for criminal prosecution purposes.

1.2 Background

According to Smita N. and Dharaskar V. [1.2], the authorship analysis research field can be separated into three different sub branches, which are the authorship identification, authorship characterization and similarity detection. Each branch serves for a different purpose in the authorship analysis research. All of them will be briefly introduced in the Analysis categories section [1.2.1]. This brief introduction about the authorship analysis helps the readers to capture an overview about this research area and all of its sub branches.

As the requirements from the Kalooga company, which refer to the problems in the authorship identification field, most of the features and techniques that are going to be introduced in this thesis will be corresponding to the authorship identification field. The vision of the authorship identification is to seek for patterns that people left in their writings, then classifying different people bases on their writing patterns. In order to search for hidden patterns in each writing, a tool called feature extractor, which contains a set of feature finding rules, will be used to find the writing patterns in a given text. The basic set of rules that is currently used in the feature extractor from the existing researches will be introduced in the Analysis features section [1.2.2]. Besides the feature extracting tool, in order to distinguish different persons by using the extracted patterns from their writings, a classifier will be used. The Analysis techniques section [1.2.3] will briefly introduce different types of classifier that are used, along with a quick comparison of the introduced classifiers in term of their performance.

In this thesis, we focus on online texts, messages such as online blogs, news articles or emails that contain more than 250 words. Unlike the authorship identification tasks in the literary works and published articles, which do not have the constraint in the limited length in the number of words, the online texts and messages are usually limited in their word length. Forsyth and Holmes [1.9] claimed that it was extremely difficult to attribute a writing that has less than 300 words to its original writer. The limitation in the word length of a message causes problems to some features that can only work well in the regular writings that have sufficient length. The reason belongs to the vocabularies that are used in short writings, which are relatively limited and usually unstable. This limitation in word length makes some measurement techniques such as vocabulary richness calculation becomes ineffective. Thus in this thesis, we only focus on online texts and messages that have more than 250 words.

1.2.1 Analysis categories

In the authorship analysis research area, there are three different sub research branches, and each one serves for a different purpose. The three sub branches are the authorship identification, authorship characterization and similarity detection.

Authorship identification (a.k.a authorship attribution) is used to determine the probability that a piece of writing was produced by a particular person by examining the other writings from that person.

Authorship characterization is a technique for determining the personal attributes of an author such as the gender, age, education level or the culture background by using existing writings from that author.

Similarity detection compares different pieces of writing and determines if they were produced by the same person or not. The most popular usage of this technique is the plagiarism detection.

In this thesis, as said, the main focus is on the authorship identification area since the main objective is to determine the authorship of online texts and messages. The authorship characterization research will be mentioned later in the future discussion section [VII] as an opportunity to enhance the performance of the solution for solving the authorship identification problem.

1.2.2 Analysis features

In the authorship identification research area, in order to distinguish different persons by using their writings, the stylistic characteristic in their works must be extracted for further analysis. There are four different categories of the writing style features that can facilitate the authorship identification research, which are the lexical features, structural features, content specific features and the syntactic features. All of the mentioned features in the above four categories are the sub elements of the stylometric features category.

Lexical features are supported by words and characters based analysis. With the word-based analysis, the characteristic of a person can be told by the vocabulary richness in their writings, combined to the other measurements such as the average number of words per sentence or the total number of words in a text. With the character based analysis, some features such as the total number of characters, the average number of the characters per sentence, the characters per word or the frequency in usage of the individual letters helps to reveal the writing habits of a particular person.

Structural features, which focus on analyzing the layout and organization of the text. Structural features come to play an important role when the writing's length is not sufficient for other analysis features in order to distinguish the differences among the writings. According to O.De Vel [1.3], structural features are particularly important when analyzing short writings such as emails and online messages since they refer to the regular tendencies in how a person constructs their paragraphs, such as the total number of sentences that starts with lowercase or uppercase letters, or in the habit of using greeting and farewell words.

Syntactic features, which focus on searching and analyzing the writing patterns that used to construct the sentences. Syntactic features focus on analyzing the usage of common words in English such as the punctuation and function words, in order to distinguish different persons by basing on the discriminating power of the syntactic features since they are derived from the people's different habits in organizing the sentences.

Content specific features refer to words that are relevant in a particular topic domain. In this thesis, this feature category is not in use. The reason is we want to examine how the authorship identification techniques and other authorship identification features can perform in general cases, which do not depend on the content of the given writings.

1.2.3 Analysis techniques

In the early state of the authorship identification research, the statistical univariate methods were the mainly used analytical tools [1.2]. Some famous tools that were used in the works of other researchers can be listed as Naive Bayes classifier (probabilistic learning and classification), B.CUSUM (Cumulative sum control chart), statistic procedure (sequential analysis technique for monitoring change detection) and cluster analysis (exploratory data for sorting them to clusters) methods. Although they did make good progress in helping to solve the authorship analysis problem with a high level of accuracy [1.4], they still have some weaknesses when comparing to the modern techniques such as machine learning. For examples, Holmes [1.1] discovered that some statistical univariate methods such as the CUSUM technique is unreliable since the stability of the tests that spread over multiple topics was not guaranteed. Moreover, the statistical univariate methods can only deal with a small number of features, or the pitfalls problem, which requires more stringent in assumptions and models [1.5].

When the computing power has increased over the last few years, the machine learning techniques emerged. Some of available machine learning techniques are being used in the authorship identification research can be named as the support vector machine, neural network and the decision tree. In recent years, these machine learning techniques have been widely accepted since they can provide a better scalability compare to the statistical univariate techniques in term of handling more features, and less susceptible to noisy data [1.6].

In general, besides the difference in the number of features that can be used between the statistical univariate and the machine learning techniques, the accuracy level can be achieved higher when using the machine learning techniques than when using the mathematical statistical analysis techniques [1.6]. The machine learning techniques can handle a larger set of features with fewer requirements on the mathematical models and assumptions [1.7]. Furthermore, the machine learning techniques are also tolerant to the noise and nonlinear interactions among features [1.8].

1.3 Problem

One of the main purposes of this thesis is to examine the existing features and techniques that are currently used in the authorship identification research area. This examination helps to find the weaknesses in the current features and techniques, which can be used later to seek for the possible improvements.

This goal leads to the following research questions:
a. What are the current existing features and techniques that are being used in the authorship identification research field?

b. How can the existing features and techniques be compared in terms of their qualitative (accuracy level) and quantitative (number of the different person they can be distinguished and the execution time of the solution in performing that task) attributes?

c. What are the possible ways that can help the existing techniques to improve their performance in term of qualitative and quantitative attributes?

In the first part of this thesis, the main problem will be seeking for the existing features and techniques that can be used for solving the authorship identification problem, the biggest concern here is choosing for the appropriate feature sets and classifying techniques, in order to implement them into a completely working prototype.

In the second part of this thesis, we will focus on how to improve the existing features and techniques. The aim of this work is to improve the qualitative (accuracy level) and quantitative (performance ability) attributes of the implemented prototype in the first part.

1.4 Solution

In the first part, which is seeking for the existing features, techniques and implementing them into a prototype for solving the authorship identification problem, the research works from different researchers in this field will be examined and evaluated for their qualities. Then a prototype will be implemented bases on the existing features and techniques that are considered to have reasonable qualities. The chosen features and techniques can come from an individual researcher or can be a combination from the works of multiple researchers if it is necessary in helping to improve the quality of the developing prototype.

In the second part, which focuses on improving the qualitative and quantitative attributes of the developed prototype, the chosen features and techniques will be taken into account for further investigations in order to find their weaknesses. From that, possible improvements will be proposed in order to increase the qualitative and quantitative performances of the prototype. All of the proposed improvements will be implemented to the initial prototype and examined for their working abilities.

1.5 Contribution and organization of the research

This thesis attempts to look for an outstanding candidate in the available existing features and techniques in the authorship identification research area. This attempt is done by choosing good quality features and techniques from the previous researches and examining them against each other by comparing their qualitative and quantitative performances in order to choose the best candidate. After that, the chosen candidate will be taken for further investigations to seek for its weaknesses in term of its performances in qualitative and quantitative attributes. From that, one or more possible improvements will be proposed and examined for their likelihood of success.

II. State of the art

2.1 General procedure

Before going deeper to the current state of the authorship identification research area, the general procedure that is commonly used by other researchers when solving the authorship identification problems will be introduced. This introduction helps the readers to have a better view on the differences among the existing researches, which will be shortly introduced in the later section [2.2]. The classic approach to model a solution for solving the authorship identification problems is a combination of four different ordered steps.

Step 1: Finding the data collection.
In this step, the text data that will be used for the authorship identification purpose must be collected. Data can come from various sources, and mostly depends on the purpose of the researchers. Data can be posts from a discussion forum, a set of emails or a collection of blog articles.

Step 2: Feature extracting on the collected data.
In this step, bases on the information that the researchers want to search from the data collection, each sample in the data collection will be processed in order to extract their writing patterns by using the feature extractor which built from a predefined set of stylometric features.

Step 3: Generating the model.
Up to this step, the data collection will be divided into two different smaller subsets. The first subset is called the training set, which contains the samples that will be used for training the model, which created by the classifier. The second subset is called the testing set, which contains samples that will be used later for the testing process. The trained model that was created before by the classifier will contain person's unique extracted writing patterns from the samples in the training set.

Step 4: Identifying the original writer.
In this step, each sample in the testing set will be analyzed by the created model from the classifier for predicting their original writer.

These four steps are repeatedly used in different research works, that will be shortly introduced in the upcoming section [2.2]. In the later sections [2.3 and 2.4], the two important things that can be observed from the previous research works, which are the set of features and the model generator (a.k.a the classifier) will be presented.

2.2 Previous researches

In the authorship identification research, different aspects can affect the qualitative (accuracy level) and quantitative (performance time) values of the author identification task. Some of the highlighted aspects can be named as the amount of different writers that needs to be distinguished, the number of available messages for the training process or types of the features and techniques that will be used. The researchers must make their decisions on what they want to use before the experiments are performed. Since the researchers can freely pick which datasets they want to use, which techniques they will apply in their experiments, thus the qualitative and quantitative results from different researches become difficult to be compared.

The experiment setups from the other research works

Research paper	Total no. Persons (P)	Total no. Messages (M)	Avg. Message length (word)	Avg. M/ P
Corney et al.[2.1]	4	253	92	64
De Vel [2.2]	3	156	295	52
Zheng et al. [2.3]	20	960	169	48
Stamatatos[2.4]	10	300	1122	30
Tsuboi [2.5]	3	4961	112	1653

As a consequence, in this thesis, we will not focus on comparing which researchers have better chosen features or techniques than other researchers in performing the experiments. Instead, we will put more effort on finding how can we combine the chosen features and techniques from the different researchers for our experiments.

Since, in this paper, we focus on the authorship identification task for online texts, thus our main focus will be on the research papers that are taken into account of making use of online texts such as the blogs, news articles or email messages for the authorship identification task.

2.3 Usage of stylometric features

In order to extract the writing style of a person from their writings, a stylometric feature set will be predefined and then using by a feature extracting tool. The stylometric feature set can be considered as the most effective method for searching patterns in writings. Besides the traditional stylometric features that have been used from time to time by various researchers such as calculating the average number of sentences in a text or calculating the average number of words in sentences, some of the researchers also make use of their own additional features which they think those features can be helpful for their researches. These additional features can be very heterogeneous, from counting the number of ellipsis [2.6] to using an URL as the signature [2.3]. This is the main reason why the number of used stylometric features is different among various researches. Before going any further to look at some research papers, we want to emphasize that the additional features which are used by different researchers may work well on their researches, but may not work well on the other's researches. Their effectiveness also depends on the context of what they are used for. We should not expect that a set of features that works well on an email set will also work well on a new articles set.

Since there are four different feature categories in the stylometric features set, thus in the upcoming paragraph, we will focus on the effectiveness of each feature category in the authorship identification task. This helps us to decide which feature categories we should use and should not. All of the research papers that are going to be presented have their focuses on the authorship identification task for online texts and messages.

The first research paper is from Zheng et al. [2.3], which provides a completed framework for solving the authorship identification problem for online messages. In their experiments, the researchers made use of the completed set of all stylometric features, which are the lexical features (F1) with 87 features, the syntactic features (F2) with 157 features, the structural features (F3) with 13 features and the content specific feature (F4) with 10 features. For the content specific features, Zheng focused on the adult content words, which was corresponding to his research purpose and the dataset he was using. According to Zheng, the accuracy level in his experiments was improved when those features were combined and used together. In the end, the total number of features that Zheng used in order to extract the writing patterns from the dataset was 270 features. The below table shows the improvement in the result of Zheng's experiments when the different categories of stylometric features were combined and used.

Evaluating the different combinations of features by Zheng

Combination	F1	F1+F2	F1+F2+F3	F1+F2+F3+F4
Accuracy level	89.36%	90.03%	94.66%	97.69%

Moving on to another research paper from Nachune, R. Chandramouli and K.P. Subbalakshmi [2.6], which focused on the online news articles. In this research, instead of using all four categories in the stylometric features set, they only used three categories, which are the lexical, structural and the syntactic categories. In total, they had 153 stylometric features (130 lexical features (F1), 10 syntactic features (F2), 12 structural features (F3)), and they did not have any content specific features. Instead, they used function words (391 words) as features (F4) in order to extract the writing patterns from the writings. In the experiments, they also performed the accuracy test as Zheng et al. [2.3] did for different feature categories. One thing was different from the Zheng's research is that in this experiment, each category of features was tested individually, then all of them were combined and tested again. Interestingly, the accuracy level was also linear to the number of the features they used. The below table shows the improvement in the result of the experiment when the different feature categories were combined and used.

Evaluating the combinations of features by Nachune

Combination	F1	F2	F3	F4	F1+F2+F3+F4
Accuracy level	66%	65.37%	61.26%	74.81%	85.13%

Another research paper is from A. Abbasi and H. Chen [2.7]. Similar to Zheng et al. [2.3], Abbasi and Chen also used the lexical, syntactic, structural and the content specific feature categories in their experiments. With the same argument as Zheng, they also proved that the accuracy level of the authorship identification task could be also improved if different categories of features were combined and used. In total, they used approximately 234 different features for extracting the patterns from the dataset in their experiments. The below table shows the improvement in the result of their experiments when different categorizes of feature were combined.

Evaluating the combiantions of features by Abbasi and Chen

Combination	F1	F1+F2	F1+F2+F3	F1+F2+F3+F4
Accuracy level	88.00%	90.773%	96.50%	97.50%

In general, the stylometric features set is heavily used by the researchers in the authorship identification research field. Up to now, it is still the most effective way in order to reveal the writing patterns of a person from their writings. The stylometric features become most effective when their different feature categories are combined and used together. Besides the features from the lexical, structural and syntactic categories, which mostly remain the same in different research papers, the content specific feature category has an exclusive focusing on the context of the researching purpose, such as looking for online child abusing or scams by searching for the content specific words in the dataset. In this thesis work, we focus on the general cases of online texts and do not tie to any specific topic. Thus, the content specific features are out of the researching scope of this thesis.

2.4 Usage of machine learning algorithms

There are two types of machine learning algorithm, the supervised and unsupervised learnings. For the classifying tasks such as the authorship identification, which requires algorithms that can find the differences among the input data by using their attributes, the supervised learning algorithms are the perfect choice for that purpose. Moreover, the supervised learning algorithms are also suitable when the information of both training and testing data (a.k.a the data's label) is known beforehand. Thus in the classification process of the authorship identification task, the supervised learning algorithms are mainly used.

Unlike the supervised learning algorithms, the unsupervised learning algorithms do not need the information of the data beforehand. Unsupervised learning algorithms use the extracted characteristics of the samples in order to group them into different groups that sharing the similar properties. The unsupervised learning algorithms will be used later as a method for finding possible improvements for the authorship identification task.

2.4.1 Supervised learning algorithms

Each sample in the training dataset will be firstly processed by a feature extractor, and resulting in a labeled characteristic list (a.k.a writing patterns) for all sample data. This task is done by processing the raw samples with the predefined stylometric feature set in the feature extractor. This process is also applied to the samples in the testing dataset. After the feature extractor processes all of the samples in the dataset, the supervised learning algorithms will now learn the extracted characteristic list from the samples in the training dataset and then creating a model. This model will be used to generalize over the processed samples of the testing dataset. It will predict the label for each sample in the testing dataset, bases on what it learned from the training dataset. This predicting process is one of the important steps in the authorship identification task. Currently, there is a few different types of supervised learning algorithms that can help to achieve this task.

Result from the authorship identification task depends a lot on the types of supervised learning algorithms that are used. Some of favored machine learning algorithms that are used in the authorship identification research field are the neural network, the support vector machine and the decision tree. The below table will give to the readers a perspective about the popularity in the usage of each introduced supervised learning algorithm in the authorship identification research area.

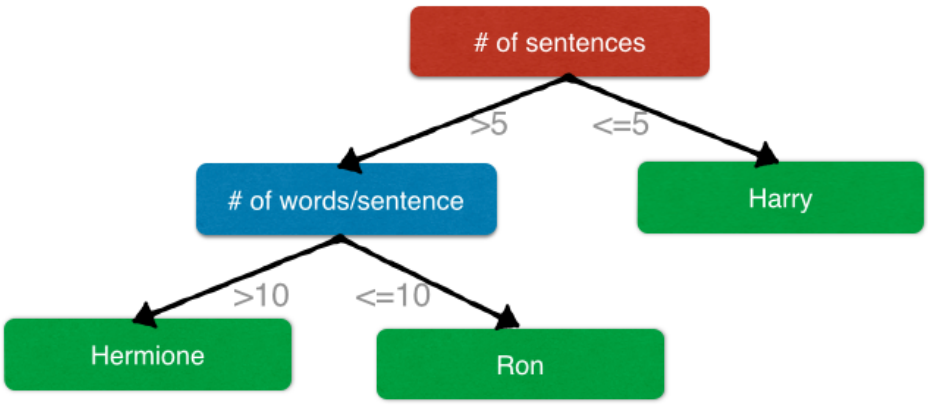
Used classifiers from other research works

Research paper	Decision tree	Neural network	SVM
Corney et al. [2.1]			v
Graham et al. [2.8]		v	
De Vel. [2.2]			v
Zheng et al. [2.3]		v	v
Kiell [2.9]		v	
Tsuboi [2.5]			v
Ahmed and Hsinchun [2.7]	v		v
Hoorn [2.10]		v	

In the upcoming section, brief descriptions about the supervised learning algorithms such as the decision tree, the neural network and the support vector machine will be introduced. This brief introduction will help the readers to have the information about the current supervised learning techniques are being used in the authorship identification research area, also provides the information about how these supervised learning algorithms can be used for the classification (authorship identifying) task later.

2.4.1.1 Decision tree

Decision tree is a decision supporting tool that uses a tree-like model or graph to organize the decisions and their possible consequences. There are three main components to form a decision tree, which are the root node, the chance node and the end node. In the decision tree technique, a tree structure will be used to model the extracted characteristics of the samples in the training dataset. The root node represents for a feature and it will connect to other chance nodes, which represent different decisions when evaluating the feature. Each chance node can be a new root node, which represents for another feature, or can be an end node, which contains the result for the classification task. In a decision tree, each testing sample will need to go through different routes of root nodes and chance nodes in order to get to the final end node, which represents for the label that the testing sample should belong to. A compact example of how a decision may look is demonstrated in the below figure.



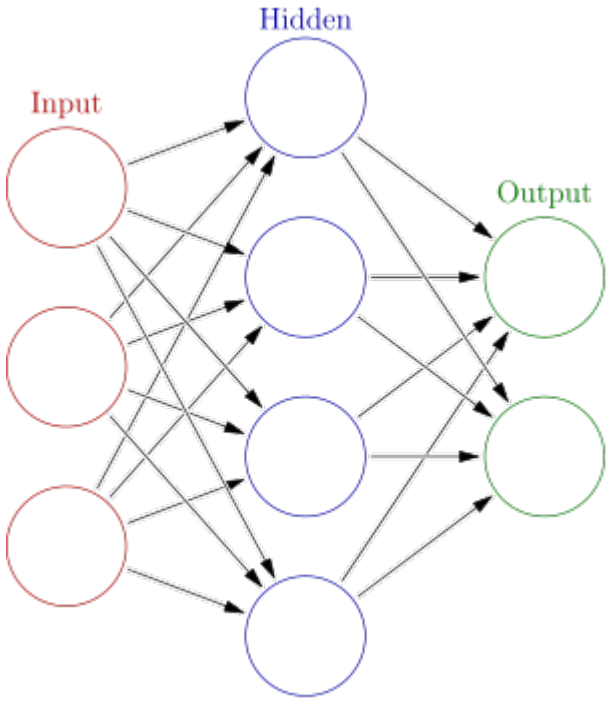
Decision flow in a decision tree

The most advantage of the decision tree is that once the tree is constructed (formalized model), the classification process (Step 4 in the introduced general procedure) in the authorship identification task can be performed very fast. Moreover, when there is a few highly correlated features, whenever one of them has been chosen to be used during the classification process, the other ones will not be chosen anymore.

In the authorship identification research area, most of the extracted values from the feature extraction process are numeric. In order to build a decision tree, these extracted values must be splitted into different categories (number of sentence > or <= 10 in the above example). The problem is that this categorizing process is computationally expensive, but it is a crucial factor for the success of the built decision tree. C4.5 is a popular representative for the decision tree algorithm which is used in the authorship identification research field. C4.5 was developed by Quinlan(2.11) and Weka (2.12) and tested in Zheng et al.[2.3] research for its accuracy performance, compares to the neural network and machine learning techniques. The result shows that the performance of the C4.5 technique is the lowest compares to the results from the other two techniques. The result from Zheng's paper is also tolerant to the result from Ahmed and Hsinchun's research paper [2.7]. Ahmed and Hsinchun performed a experiment between the C4.5 and the Support vector machine technique, the result shows that the accuracy performance of the C4.5 technique was significantly lower than the result from the support vector machine technique, with the same usage of the combination of all the stylometric features categories, the C4.5 technique only scored 71.93% on the accuracy level experiment compares to 94.83% of the support vector machine technique in the same dataset.

2.4.1.2 Neural network

The neural network is a computational model for pattern recognition. Neural network is inspired by the animal's central nervous system. A neural network is a system of the interconnected neurons by the directed weighted links between them, and each neutron has its computational power. There are three main layers in a neural network, which are constructed by a numerous of neurons in each layer. The first layer is called the input layer, and this layer represents for the stylometric features. The second layer in a neural network is called the hidden layer. A neural network can have multiple hidden layers. The hidden layer is responsible for processing the inputted value from the input layer. The final layer is called the output layer. This layer is used to present the output value after the model completes its computational process. This output value is the label of the sample in the training set. An example of a simple neural network is demonstrated in the below figure.



An example of a neural network's inside structure.
Source: Wikipedia

When a neural network is just created, each value of the input node is given with a random weighted sum value, which then will be processed by an activation function in the hidden layer in order to determine the output of that node. Before the neural network can be used, it needs to be run through a learning process, which helps to adjust the weight of the network until the learning error rate is minimized to an acceptable value (usually around 0.01% error rate).

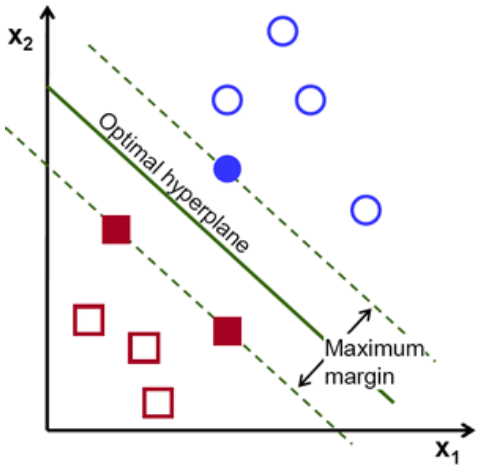
Setting up a neural network is rather complex since there is a lot of parameters that need to be configured before the learning process can start. Besides the easily configurable parameters such as the number of neurons in the input layer (actually the number of used features in the feature extraction process), or the number of neurons in the output layer, which is the number of possible outputs from the dataset, other parameters such as the number of hidden layers, the number of neurons in each hidden layer or the activation function that should be used is quite complicating to set. Inappropriately settings in these parameters may result in the under-fitting problem, when the neural network cannot give a clear distinguish among the training samples, or even the overfitting problem, when the neural network can fit the data in the training set well, but fail to generalize the data in the testing set.

2.4.1.3 Support vector machine

The support vector machine, also known as the support vector network, is a supervised learning algorithm that is used for classification and regression analysis. In this thesis work, we focus on the classification ability of the support vector machine technique, the regression analysis ability is out scope since it focuses on predicting the continuous data, which is irrelevant to this research.

When a model is created by the support vector machine classifier with the extracted characteristic data from the training set, the support vector machine technique will try to look for the optimal hyperplane with the maximum margin, which can separate the training data into different sets. This process of finding the hyperplane bases on the structural risk minimization, which helps to minimize the generalization error while minimizing the training error rate and try to keep the trained model is not becoming too complex. The bigger the margin of the hyperplane is, the smaller the generalization error will be. A small hyperplane margin is prone to overfitting the data, which makes the created model may fail in generalizing on the testing data later.

The below figure shows a basic example of an optimal hyperplane. In this case, only data points that help to determine the largest space of the margin are considered, together, they form the support vectors (Points located on the dotted line).



An example of how the support vector machine technique uses hyperplane to distinguish two different classes.
Source: Wikipedia

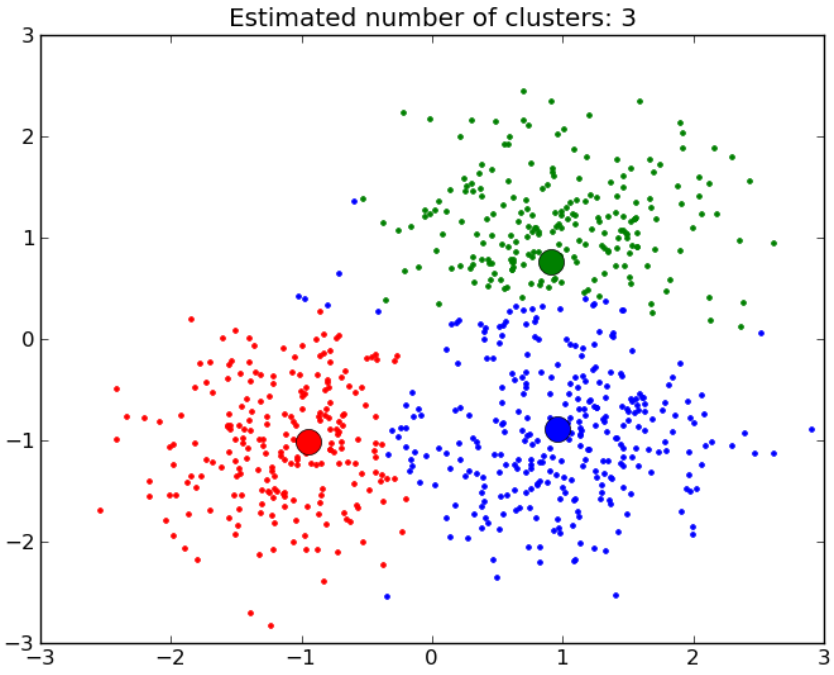
Note that in the above example, the demonstrated hyperplane is not the only way to separate the data. The data separation task also depends on the kernel that was chosen during creating the model (training process). Some popular kernels in the support vector machine technique are the sigmoid kernel, the radial basis function kernel and the polynomial kernel.

The support vector machine technique does not only limited support to separate two different classes, it can also work with a larger amount of classes, which is helpful in the authorship identification research area, where a writing may come from not two, but various persons. In order to do this, the support vector machine technique uses the pair wise classification method. This method works as it will create a classifier for each pair of classes, the data that does not belong to these two classes will be ignored.

The previous research works showed that the support vector machine technique can give higher accuracy result compares to other techniques. Zheng et al. [2.3]'s experiments pointed out that the result from the accuracy performance of the support vector machine technique is higher than the result from the C4.5 and the neural network techniques. The experiments from Ahmed and Hsinchen [2.7] also showed the same result.

2.4.2 Unsupervised learning algorithms

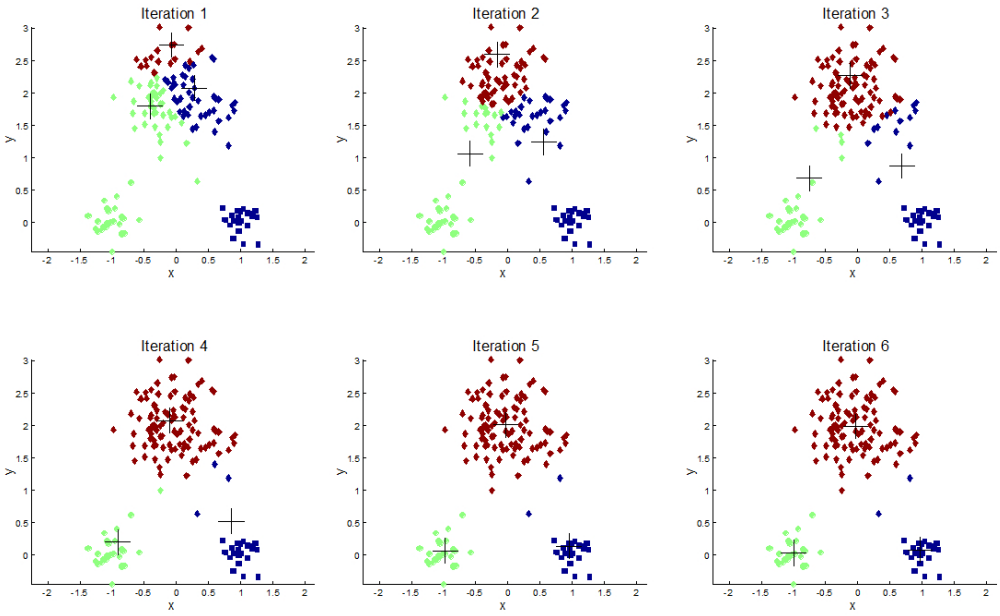
Different to the supervised learning algorithms, where the information about the label of the training data must be known beforehand, the unsupervised learning algorithms try to look for the hidden structures in the unlabeled data. Some approaches that are used in the unsupervised learning algorithms include the clustering ability, the hidden Markov models and the blind signal separation. In this thesis work, we only focus on clustering approach, which is the main task of the exploratory data mining. Clustering process helps to group different data into groups (a.k.a clusters). Each group will only contain data that is similar to each other. Comparing to the supervised learning algorithm which is an essential technique in the authorship identification research area, the unsupervised learning algorithms are not popularly used in this research area because they do not provide what the researchers need, which is the classifying ability. In this thesis work, we will focus on using the unsupervised learning algorithm as an external tool to see how it can help to improve the accuracy level and the performance time in the authorship identification task compares to the traditional techniques. According to O.A. Abbas[2.16], some of the popularly used unsupervised learning techniques are the k-means, the hierarchical and the self-organized map techniques. The below figure is an example of a successful clustering process in order to organize the unlabeled data.



An example of a successfully clustering process.
Source: Wikipedia

2.4.2.1 K-mean

The K-mean algorithm was introduced by MacQueen, 1967 [2.13] and were one of the simplest unsupervised learning algorithms which can be used to solve the clustering problems. The clustering process follows a simple procedure to classify any given dataset. The K-mean algorithm requires that the number of produced clusters must be manually given before the clustering process. The idea is that firstly, the k-centroid values will be defined, one for each future cluster. Next, each sample in the dataset, which is represented by their extracted characteristic from the feature extracting process, will be linked to the nearest centroid. When this linking process is completed, the K-centroids will be repeatedly re-calculated until the locations of these k-centroid are optimal. At this step, the clustering process is completed in separating the given data into the clusters that represent the different groups of data, which share the similar attributes. The below figure demonstrates the loop in calculating the centroids' value during the clustering process.



An example of how k centroids are located in the K-mean technique.
Source: Wikipedia

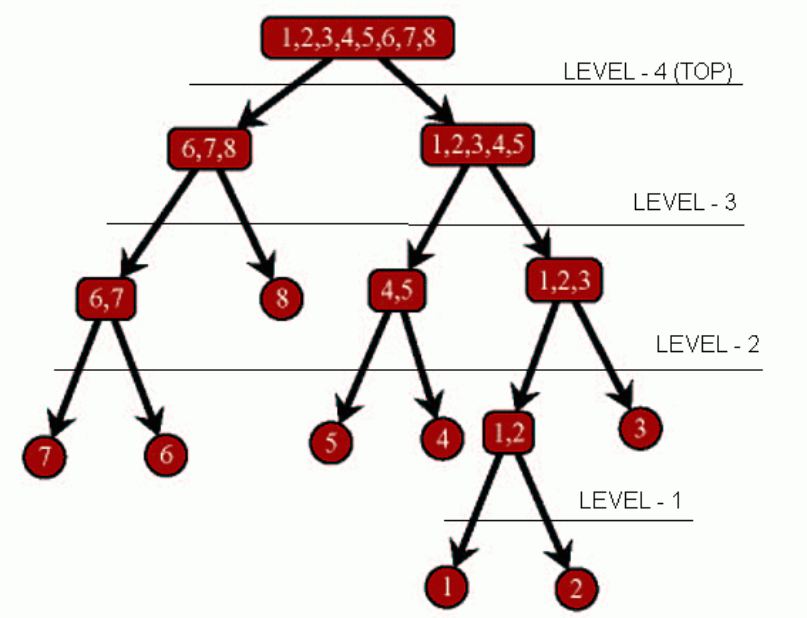
2.4.2.2 Self-organizing map

The self-organizing map (SOM) algorithm, also known as the Kohonen network, is a type of the artificial neural network. The difference between the SOM technique and the other artificial neural networks is that the SOM algorithm uses a neighborhood function to preserve the topological properties of the input space. This usage of the neighborhood function helps the SOM algorithm become useful in multidimensional scaling (The level of the similarity of each sample in the dataset), which will become useful in the later part of this thesis work.

A self-organizing map consists of multiple neurons and weight vectors; each neutron will associate with its weight vector. This weight vector will have the same dimensions to the input data vectors, which was created before, during the mapping process when the input data is processed by the algorithm. Every neuron in the network will be examined for its weight value. Neurons that have weight values that are more similar to the input data vectors are called best matching units. After that, the radius of the neighborhood of the best matching unit will be calculated. Any neuron that is found within the calculated radius will be adjusted in order to make them more like the input data vectors. The further a neuron is from the best matching units, the less its weight is altered. [2.14]

2.4.2.3 Hirachical

The hierarchical clustering algorithm aims to build a hierarchy of clusters. This process can be done in two different ways, the agglomerative way and the divisive way. In the agglomerative approach, in the beginning, each sample in the dataset will stay in its cluster. Every time the hierarchy goes up for one level, pairs of clusters are merged. In the divisive approach, in the beginning, all of the samples in the dataset will stay in a same cluster together. When the hierarchy moves down for one level, the current clusters will be splitted into the smaller clusters. Both of the agglomerative and divisive approaches make the merging or splitting decisions base on the measurement value of the dissimilarity between the sets of observations. This can be achieved by using an appropriate metric and a linkage criterion [2.15]. In the agglomerative approach, the clustering process is completed when all samples stays in the same cluster, while, in the divisive approach, each sample will stay in its cluster. Below figure demonstrates a simple hierarchy result after the clustering process with the divisive approach.



An example of a successful clustering process by the Hirachical technique.
Source: Wikipedia

III. Design

As a part in this thesis work, which is implementing a prototype bases on the existing solutions from the authorship identification research area, and examining the prototype for its qualitative (the accuracy level) and quantitative (the performance time) attributes. Therefore, in this chapter, the researching works from the other researchers, which were represented in the second chapter [II], will be examined and evaluated in order to make decisions on which features and techniques we want to apply to our prototype.

3.1 Features

Since we focus on implementing a prototype that can work independently from the contents of the data which will be used in the experiment, the content specific feature category will not be used in the implemented prototype. For the other three remaining feature categories, we will adapt the English feature sets from the previous authorship studies, which were presented in the second chapter. Our feature set will consist of 465 features, including 89 lexical, 06 structural and 370 syntactic features.

The lexical features (L)

L1 to L6	Total number of (characters, alphabetic characters, upper case characters, digit characters, white space characters and tab spaces).
L7 to L32	Frequency of alphabetic characters A-Z (not case sensitive).
L33 to L53	Frequency of special characters (~, @, #, $, %, &, *, (,), -, _, =, +, >, <, {, }, |, \\\\, /, ?).
L54 to L56	Total number of (words, short word < 4 characters, characters in words).
L57 to 59	Average of (word length, sentence length in term of character and in term of words).
L60	Total different words
L61	Hapax legomena (Frequency of once-occurring words)
L62	Hapax dislegomena (Frequency of twice-occurring words)
L63	Measuring vocabulary richness by Yule's measurement [3.1]
L64	Measuring vocabulary richness by Entropy measurement [3.1]
L65	Measuring vocabulary richness by Simpson's D measurement [3.1]
L66	Measuring vocabulary richness by Sichels S measurement [3.1]
L67	Measuring vocabulary richness by Honores measurement [3.1]
L68 to 89	Word-length frequency distribution (from 1 to 21 characters).

The structural features (R)

R1 to R4	Total number of (lines, sentences, sentences begin with upper/lower case letters).
R5	Start with greeting ("hi", "hey", "hello", "dear")
R6	End with farewell ("see you", "care", "love", "take")

The syntactic features (S)

S1-9	Frequency of punctuations (",", ".", "?", "!", ":", ";", " ' ", "\"")
S10 - S370	all #another #any #anybody #anyone #anything #both #each #each other #either #everybody #his #everyone #I #everything #it #few #its #he #itself #her #little #hers #many #herself #me #him #mine #himself #more #most #much #myself #neither #no #one #nobody #none #nothing one #one #another #other #that #what #others #theirs #whatever #ours #them #which #ourselves #themselves #whichever #several #these #who #she #they #whoever #some #this #whom #somebody #those #whomever #someone #us #whose #something #we #you #are #can #aren't #cannot #ain't #can't #'re #could #be #couldn't #been #did #didn't #hadn't #haven't #do #'d #'ve #don't #has #is #does #hasn't #isn't #doesn't #'s #'s #had #have #may #might #shouldn't #mightn't #was #mustn't #wasn't #shall #were #shan't #weren't #should #will #won't #'ll #would #wouldn't #'d #and #or #though #because #yet #unless #nor #so #when #now that #even though #although #if #now #that #only #if #while #whereas #whether or not #in order that #in case #even if #until #adios #ah #aha #ahem #ahoy #alack #alas #all hail #alleluia #aloha #amen #attaboy #aw #ay #bah #dear #begorra #doh #behold #duh #bejesus #eh #bingo #encore #bleep #eureka #boo #fie #bravo #gee #bye #gee whiz #cheerio #gesundheit #cheers #goodness #ciao #gosh #crikey #great #cripes #hah #Ha-ha #hail #hallelujah #heigh-ho #hello #hem #hey #hey presto #hi #hip #hmm #ho #ho hum #hot dog #howdy #hoy #huh #humph #hurray #hush #indeed #jeepers creepers #jeez #lo and behold #man #my word #now #ooh #oops #tush #ouch #tut #phew #Tutetut #phooey #ugh #pipepip #uh-huh #pooh #uh-oh #pshaw #uheuh #rats #viva #righto #voila #scat #wahoo #shoo #well #shoot #whoa #so long #whoopee #Touch #whoops #whoosh #wow #yay #yikes #yippee #yo #yoicks #yoo-hoo #yuk #yummy #zap #aboard #astride #down #of #through #about #at #during #off #throughout #above #athwart #except #on #till #absent #atop #failing #onto #to #across #barring #following #opposite #toward #after #before #for #out #towards #against #behind #from #outside #under #along #below #in #over #underneath #alongside #beneath #inside #past #unlike #amid #beside #into #per #until #amidst #besides #like #plus #up #among #between #mid #regarding #upon #amongst #beyond #minus #round #via #around #but #near #save #with #as #by #next #since #within #aslant #despite #notwithstanding #than #without #worth #according to #ahead to #as to #aside from #because of #close to #due to #except for #far from #in to #into #inside of #instead of #near to #next to #on to #onto #out from #out of #outside of #owing to #prior to #pursuant to #regardless of #subsequent to #as far as #as well as #by means of #in accordance with #in addition to #in case of #in front of #in lieu of #in place of #in spite of #on account of #on behalf of #on top of #versus #concerning #considering #regarding #apart from

3.2 Supervised machine learning algorithms

From the performed comparison on the popularity of the different supervised machine learning algorithms in the previous chapter [II], it is easy to see that the neural network and the support vector machine techniques are the most two popular classifiers which are currently used in the authorship identification research area. Thus in this paper, the classifier component of the prototype will be implemented with both neural network and support vector machine techniques.

In the concern of the support vector machine technique, originally it is a two classes classifier only. For some research areas such as authorship identification area, which includes variously different classes in the identifying process, two classes will just be not enough. Thus, there are two possible solutions, called the one-versus-one and the one-versus-all classifications, which can help the support vector machine technique to work with a larger number of classes, instead of two classes only. With the one-versus-all approach, it will try to find the class which classifies the test datum with the greatest margin while, with the one-versus-one approach, it tries to combine the binary classifiers to solve the multi classes problem. According to C. H. Hsu and C. J. Lin, IEEE 2002 [3.2], from their research paper, in practice, the one-versus-one approach is more suitable than the one versus all approach, which can give the same accuracy level while offering a better performance.

In the neural network technique, currently there are more than five approaches to formalizing a neural network. Some popular ones are the feedforward neural technique and the radial basis function networks technique. According to the research of B. Wilamowski [3.3], the feedforward neural network approach is more efficient and can get better accuracy result in the classification task, compares to the radial function network approach. Thus, in this paper, which mainly focuses on the classification task, the feedforward network approach will be used for the prototype.

Another point to be considered in the neural network technique is about the learning approaches. Currently, in a feedforward neural network, there are three main learning approaches, which are the propagation, the back propagation and the resilient propagation learning. For the prototype in this thesis work, the back propagation approach will be used. This decision is made bases on the research works from Jeff Heaton [3.4], which recommends that the resilient propagation learning approach is one of the most efficient training algorithms in the supervised feedforward networks. Moreover, this learning approach also requires no configuration for parameter settings before it can be used. This lack of needs for configuring parameters helps to avoid the difficulties in determining some parameters such as the learning rate or the momentum values like in the other learning approaches.

For both neural network and support vector machine techniques, they use the same mathematical material to solve the classification problem. In the neural network techniques, the mathematical function is called the activation function, which is attached to the layers of the networks. In the support vector machine techniques, the mathematical functions are called the kernels. Some popular activation functions in the neural network technique are the activation gaussian, the activation sigmoid and the activation linear activation functions. Some popular kernels in the support vector machine techniques are the polynomial, the radial basis function and the sigmoid function.

According to J. Heaton [3.4], the activation sigmoid function is a popular choice for the feedforward and the recurrent neural networks. The activation sigmoid function can also work well with non-negative data, which is suitable for this research since the extracted data by the feature extractor is positive data only. The interesting fact here is according to M. Mohri [3.5], when a neural network uses sigmoid as its activation function, that neural network can be considered as the ascender of a support vector machine classifier when it uses the sigmoid function as its kernel. This is an advantage when we try to compare for the qualitative and quantitative attributes between a neural network classifier and a support vector machine classifier since they will use the similar mathematical material on the same dataset, which will be presented in a later chapter [IV]. This helps us to have a fair view between these two classification techniques.

In general, a neural network demands more manual configuration than the support vector machine technique. In order to construct a neural network, the number of the hidden layers and their amount of neurons must be provided. There is no existing procedure or method that can help in deciding how many hidden layers that a neural network should have, or what is the amount neurons in the hidden layers to be considered as sufficient. On the other hand, in the support vector machine technique, the most important variable to be decided is the kernel, that will be used for the classifying process. Other variables can be properly chosen easily [2.17]. Furthermore, a notable point is that, the results from the support vector machine technique are consistent no matter how many times the classifying process are repeated. In the neural network technique, for the same parameter choosing and the same dataset, the produced results are varied. The reason is that when a neural network is created, it receives a random value for the weight value of each neutron. Then this value will be calibrated during the training process. Since the weight calibration is not a consistent process, thus after every training process, the weight of the same neutron may be different, which leads to an inconsistency in the final result.

By implementing both of the neural network and the support vector machine techniques, we have a better chance to verify the performance of the built prototype when it is used with different classifying techniques. This is also a good chance for revalidating the comparison results between the performance of the neural network and the support vector machine techniques from previous researches.

Moreover, besides the required validation in the qualitative (accuracy level) attribute of the prototype when different classifying techniques are applied, we also focus on the quantitative (performance time) attribute of the prototype when different classifiers are used. This quantitative validation will help the reader to have a better view about any tradeoff if existed, between the accuracy level and the performance time of different classifiers.

3.3 Unsupervised machine learning algorithms

From the unsupervised learning algorithms that were introduced in the previous chapter, the SOM (Self-organizing map) algorithm can be considered as the most outstanding candidate to be used in the implemented prototype since, with the advantage in multidimensional scaling, it becomes useful in the later part of this thesis. Moreover, Abbas's research work [2.17] shows that the SOM algorithm can archive in higher accuracy level in classifying most of the samples into their suitable clusters than the other algorithms.

With the hierarchical clustering algorithm, it does not scale well because of its time complexity attribute, which is O(n^2) where n is the total number of samples. The later part of this thesis work will require the unsupervised learning algorithms to be able to work on a large dataset, which makes the time complexity of the hierarchical algorithm becomes a severe disadvantage for it.

For the K mean algorithm, it requires the number of clusters must be known before the clustering process, which is inconvenient since on a randomly given dataset, we cannot predict how many possible clusters it may have. Moreover, according to Abbas [2.17], the K mean algorithm cannot handle well to the noisy data and has lower accuracy level than the other clustering algorithms such as the hierarchical and the SOM algorithms.

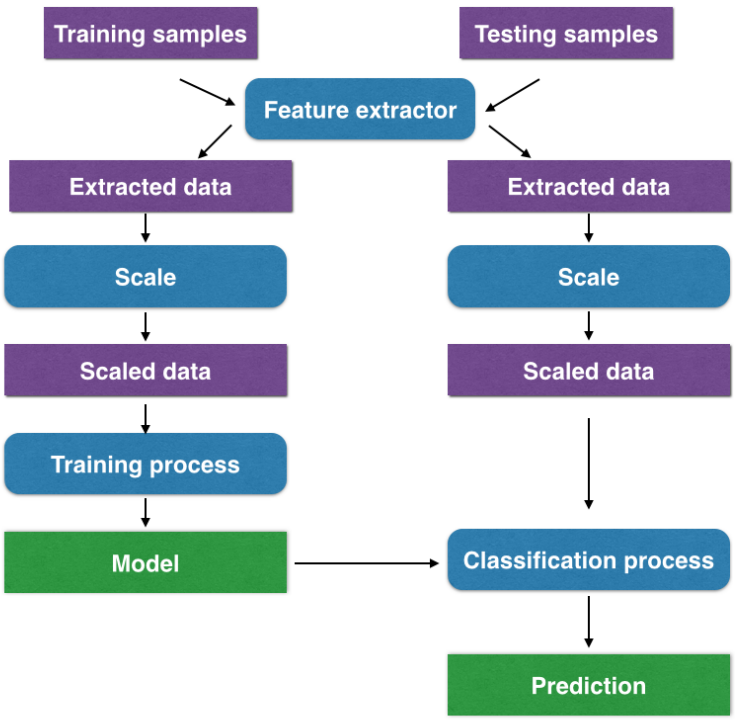
Although the SOM algorithm also has its own disadvantages, such as it cannot produce a meaningful cluster set if the data is insufficient, or it is difficult to obtain a perfect cluster set, where each cluster has its own unique value, or saying in another way, each sample appears only one time in the cluster set. Those disadvantages can be solved by some walk around methods. For examples, in the second problem, when the same sample appears in more than one cluster, we can easily filter it out and put it to a new cluster. This will be mentioned in more detail in the Implementation chapter [IV].

3.4 Design

In this section, the main functionalities of the prototype will be introduced. The Initial solution section describes the basic functionalities of the prototype, which presents for the existing works from other researchers. Sections from the project Alpha to the project Delta [3.4.2 to 3.4.5] will present for the research works in this thesis that aim to improve the qualitative (the accuracy level) and the quantitative (the performance time) attributes of the prototype from the initial solution section [3.4.1].

3.4.1 Initial solution

The prototype will be firstly constructed by using the existing works from the other researchers in the authorship identification research area. The chosen features and classification techniques from the previous sections will be used in implementing a prototype. The completed prototype will be able to extract the general characteristics of the given training and testing samples from the different persons, by using a set of feature rules that were introduced in the previous section [3.1]. Then the extracted data from the training samples will be fed to the classifier for the training purpose. Before entering the training process, the extracted data must be firstly preprocessed. During this preprocessing step, the extracted data from both training and testing samples will be scaled for improving the performance of the classification process later on. The scaled data will be given to the classifier so it can learn the differences among persons, in order to distinguish them. The training result will be synthesized into a model. Later on, during the classification process, when the testing samples are inspected for their originally writers, the model will compare the extracted characteristics from the testing samples to the information it learnt after the training process, then it will make a prediction about the original writer of the testing sample. The below figure demonstrates a simple workflow of the prototype.



A demonstration for the workflow of the prototype.

For the implementation of the feature extractor, it is only a plain programming matter of how to process text. For the implementation of the classifiers, there are a few important things that need to be mentioned.

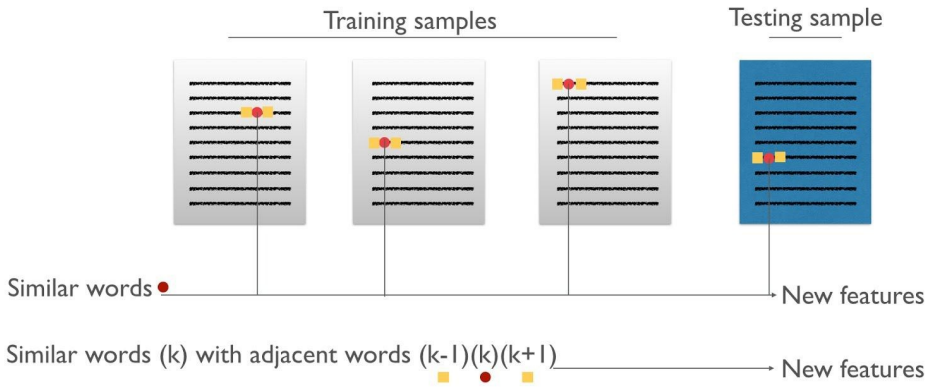
For the classifier that uses the neural network technique, as discussed from the previous section, the neural network classifier that is used in this thesis will be a feedforward neural network, with back-propagation approach as the learning algorithm and uses the sigmoid activation function in the network layers. Besides the general configurations that were already discussed, there are two more configurations that need to be determined before the neural network classifier can be constructed. Those configurations are the number of hidden layers and the number of neurons in each layer. For the number of neurons in the input and output layers of the neural network classifier, it will be decided upon the dataset is given. The number of neurons in the input layer is corresponding to the number of the stylometric features in the feature extractor, while the number of different persons in the dataset will be the number of neurons in the output layer. For the number of hidden layers and the number of neurons inside each hidden layer, there is no guide or standard procedure to choose them. According to W. S. Sarle and Cary [3.6], the situation that the performance of a neural network will be improved by adding more hidden layers is very slight, instead, most of the complex problem can be solved by just using one hidden layer. In another research work from Jett Heaton [3.4], besides proving that the number of hidden layers is not linear to the accuracy performance of the neural network, he also claimed that the optimal number of neurons in the hidden layer should be usually between the number of neurons in the input and the output layers. Bases on Sarle and Jeff's research works, the number of hidden layer in the neural network classifier will be one, while the number of neurons in the hidden layer will be the average number of the total number of neurons in both input and output layers.

For the classifier that uses the support vector machine technique, there is one more parameter that needs to be configured before the training and the classification processes, which is the cost parameter. This parameter highly depends on the dataset. The cost parameter will control the tradeoff between allowing training errors and forcing rigid margins. The cost value will create a soft margin that will permit the misclassifications. The higher value of the cost parameter is, the more chances for misclassifying points that results in a model that may not generalize well. Fortunately, the value of the cost parameter does not need to be manually set, it can be automatically determined by performing the grid search and the cross validation algorithms on the given dataset.

Up to this point, all of the mandatory configurations for reconstructing a fully working prototype from the previous works of the other researchers have been provided. In the next chapter, the necessary information for building the prototype will be described and explained in detail.

3.4.2 Improvement for the initial solution - Project Alpha

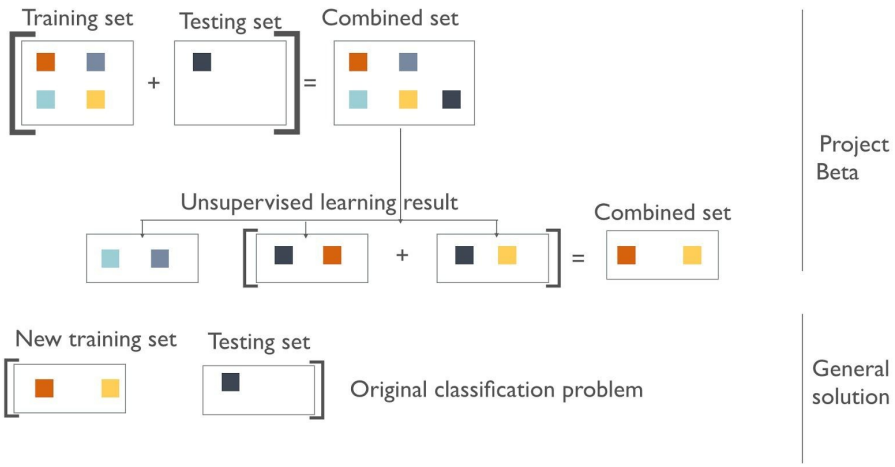
The project Alpha aims to improve the qualitative performance (the accuracy level) of the built prototype. Besides the existing stylometric features from the previous researches, we also introduce a new feature set, which is flexible in terms of size and contents, which can be dynamically changed when different datasets are used. The idea for this flexible feature set comes from the human's unconditioned reflexes, which is people tends to repeat their behaviors unconsciously. When writing, it is likely that some people may have favor in using some particular words as their habits, without even noticing about it. Bases on that idea, in the flexible feature set, we firstly find the similar words among all of the training samples of each person, then we count for their occurrences. This collected information will be added to the flexible feature set as the new features. These new flexible features change from person to person. For the better result in capturing the people's behavior, we also look at the adjacent words around the similar words, in order to seek for the people's habit in how they construct the word phrases in their writings. The below figure demonstrates for the idea in the project Alpha.



Demonstration for how project Alpha works.

3.4.3 Improvement for the initial solution - Project Beta

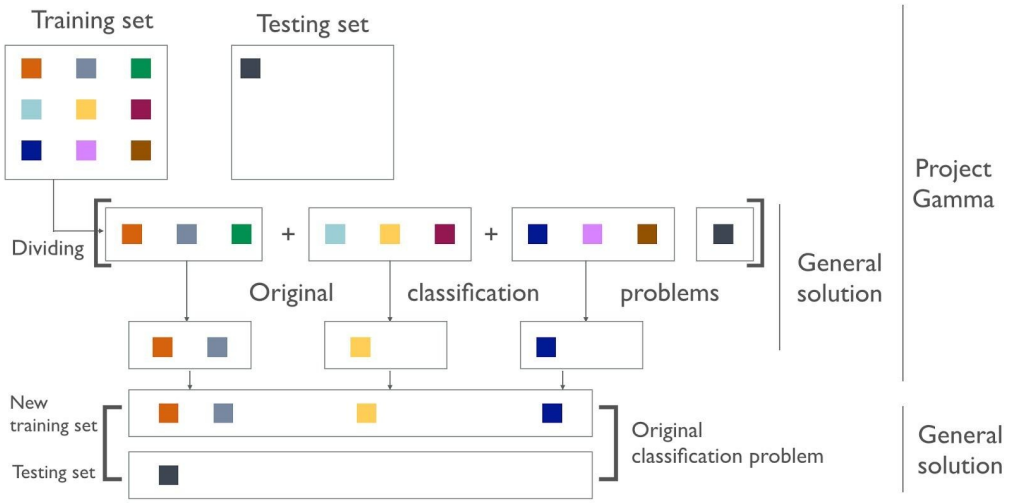
In the project Beta, the unsupervised learning technique will be applied in order to improve the quantitative (the performance time) attribute of the prototype. The idea of this project is to take advance in the ability in finding the data's similarity of the unsupervised learning techniques. In the initial solution of the authorship identification problem, when finding the original writer of a sample, all of the training data, which are samples of different persons that are suspected to be the original writers of the testing samples, will be analyzed by the classifier in order to create a model that can be used to predict for the original writers of the testing samples. The problem is that the number of different persons in the training set is linear to the execution time of the classifier in order to create the trained model, because it takes more time for the model to learn how to distinguish individual in a large group of different persons. Our aim is to reduce the number of samples in the training set, by removing the persons that have the lower chances to be the original writers of the testing samples. In order to do that, we use unsupervised learning technique to find a group of persons that share the similar writing patterns to the original author of the testing samples. The procedure is as following, firstly, the testing samples will be combined with the training samples into a combined set. In this combined set of data, testing data will be labeled as an "unknown elements", while the other training data will be labeled by their original writers' names. The unsupervised learning technique will process through all the samples in the combined set and produce clusters that contain persons who share the similarity in their writing patterns. Then, the clusters that contain the "unknown" element will be grouped together and this new group will be processed by the classification technique. All of this work is supposed to help to reduce the number of persons that needs to be classified, create a smaller training set for the classifiers, which results in increasing the performance by reducing the execution time. In this project, the worst execution time will be when the final group that is used to feed the classifier , will contains the samples of all persons in the original training set. This is the situation happens in the initial solution when the classifiers need to process through all of the persons' samples that appear in the training set. The below figure demonstrates for the idea in the project Beta.



Demonstration for how project Beta works.

3.4.4 Improvement for the initial solution - Project Gamma

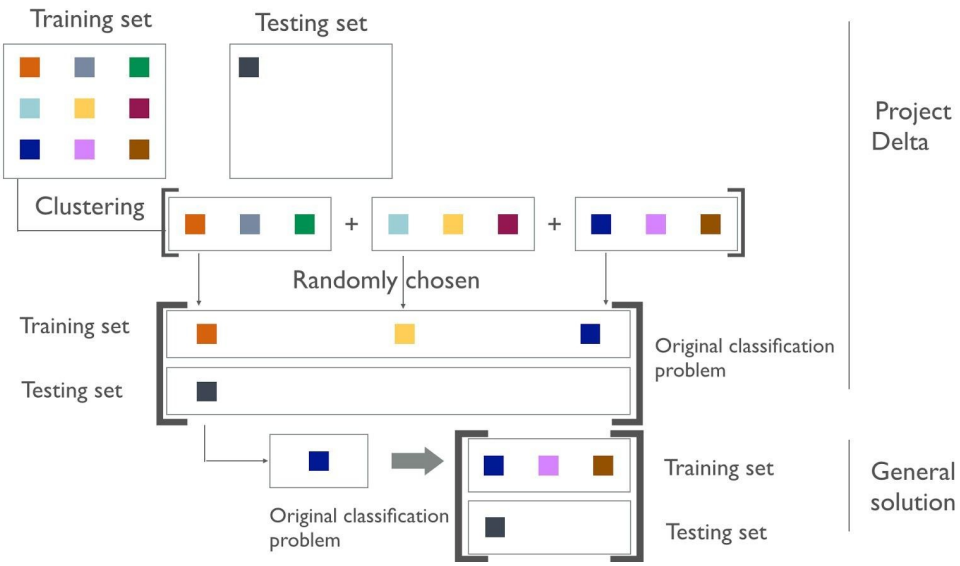
The project Gamma aims to improve the performance on both of the qualitative (the accuracy level) and the quantitative (the performance time) attributes of the initial solution. The idea for this project comes from the fact about the linear property between the size of the training set with the accuracy level and the performance time of the classification task. The bigger the training set is, the lower in the accuracy level and the higher in the execution time will be. The reason is that when there is a large number of different persons in the training set, it takes the classifier more time to create the model and the quality of the created model is not good since the classifier cannot create a perfect model which can clearly distinguish all of the given persons in the training set, no classifier can do that. Thus, in this project, we focus on solving the problem with smaller training sets. This is done as following, a large training set which contains variously different authors will be randomly divided into the smaller equal training sets, and each author will appear in one new training set only. This means when all of the divided training sets are intersected together, and the result will be empty. Then the classification process will be performed on all of the divided training datasets. In each new training set, there will be always an author that has the most similarity in their writing pattern to the writing pattern of the original author who wrote the testing sample. The classifier will be responsible for choosing those similar authors from the divided training sets. After that, all of the chosen authors from all of the divided training sets will be united into a final training set, which will be processed by the classifier again. The result from this step will be also the final result for finding the original author of the testing sample. The below figure demonstrates for the idea in the project Gamma.



Demonstration for how project Gamma works.

3.4.5 Improvement for the initial solution - Project Delta

In the project Delta, we try to make use of the unsupervised learning technique again in order to try to improve the performance time of the original solution. Unlike the project Beta, which tries to formalize the training set that contains all of the persons who have similar writing patterns with the writing pattern from the author of the testing sample. With the idea of the division in the training set from the project Gamma, in this project, we use the unsupervised learning technique and try to divide a large training set into smaller ones (clusters), each smaller training set will now contain persons that share the similarities in their writing patterns. Then one person in each cluster will be randomly chosen and united into a new group, which also contains the randomly chosen authors from the other clusters. This newly created group will be processed by the classifier to create a model. Later on, the created model will be used to predict the possible original writer of the testing samples. The predicted result will be an author from that newly created group which has their writing pattern is the most similar to the writing patterns of the original author who wrote the testing samples. This predicted author is also the person who represents for their groups (cluster), which contain other persons that also share the similarities in their writing patterns, and formed by the unsupervised learning technique from the previous processes. From this point, the cluster that contains the predicted result from the last classifying process will be processed by the classifier in order to find the person that has the writing pattern is the most similar to the writing pattern of the testing samples' author. The predicted result from this last classifying process will be also the final result for the original author of the testing samples. The below figure demonstrates for the idea in the project Delta.



Demonstration for how project Delta works.

IV. Implementation

4.1 Overview

The prototype is written in the Java programming language, using version 1.7 of the Java standard edition development kit. The prototype is divided into two main parts. The first part, which is the fundamental part of the prototype, is a completed solution for solving the authorship identification problem. This part is built by using the existing works from the other researchers in this research area. The second part of the prototype is an extension for the first part. In this second part, it will contain all of the proposed improvement projects for the prototype which were presented in the previous chapter [III], which are supposed to help the fundamental part to improves its qualitative (the accuracy level) and quantitative (the performance time) attributes in solving the authorship identification problem.

The main components in the fundamental part are the feature extractor and the classifiers. In the programming perspective, the feature extractor will act as a processor for the text data while the classifier will act as a processor for the numeric data. The data flow in the fundamental part of the prototype can be seen as following, the raw data, which contains a set of information (name) of different persons and along with their writings, will be used to feed to the feature extractor. By using the predefined set of the stylometric features, the feature extractor will now extract the writing styles of each writer by using their texts. The result from this process will be the extracted writing characteristic of each person, presented in the numeric data. After that, these numeric data will be processed by the classifier. The classifier will use the given numeric data to train a model so it can distinguish the writing styles of different persons. The result from this process will be a fully trained model, which now can be used to predict the original writers of any given writing. The feature extractor and the classifier components will be described in detail in later sections of this chapter.

The two main components in the extension part of the prototype are the flexible feature maker and the clusterer. In the programming perspective, the flexible feature maker will act as a processor for the text data while the clusterer will act as a processor for the numeric data. The main responsibility of the flexible feature extractor is looking deeper to the given texts in order to search for more writing patterns, while the clusterer responses for finding different groups of persons that share the similarities in their writing patterns. These two components will work individually or together, bases on how their functionalities are used for different purposes in the research. These two components will be used to intervene the workflow of the initial authorship identification solution, in order to change a part of the workflow or create a whole new workflow in order to improve the performance of the prototype. These interventions were explicitly described in the project Alpha, Beta, Gamma and Delta which were presented in the previous chapter [III]. The flexible feature maker and the clusterer components will be described in detail in later sections of this chapter.

4.2 Principal components

In this section, the components that perform essential processes for solving the authorship identification problem will be described. These components can be built from the existing libraries or have been developed during this research in order to serve for the experiments.

4.2.1 Feature extractor

As described before in the previous chapters, the main responsibility of the feature extractor component is to process the raw text data, using a predefined stylometric feature set in order to extract the characteristics of the given writing samples. Since we combined the feature sets from works of different researchers. Therefore, we decided to develop our feature extractor component since currently there is no existing library that can fulfill our needs. The feature extractor component will receive the information about the writers and their writing samples as inputs, the output will be a text file that contains the numeric information about the characteristics of each given writing sample and the name of their original writers, if available, encoded as numbers instead of letters so it can be understood by the classifier. The outputted text file is formatted in a way that it can be parsed by the classifier later.

4.2.2 Flexible feature maker

The flexible feature set was proposed as a possible enhancement to help increasing the accuracy level for the initial prototype that uses general solution from the existing research works. The main responsibility of the flexible feature maker will be seeking for new features, besides the predefined ones, by using data from the given writing samples for training and testing purposes. The detail about how the flexible features are created is mentioned in the project Alpha from the previous chapter [III]. The flexible feature maker receives the writing samples from both training and testing sets as inputs. The result, which produced by the flexible feature extractor component, will be a list of newly found features. This list will now be given to the feature extractor component, so the feature extractor component can use the data in the given list to update its current predefined feature set.

4.2.3 Neural network classifier

For developing a classifier that bases on the neural network technique, we decided to use existing libraries because it can help to reduce the time for developing the prototype and also minimizing the possible programming bugs in the prototype. The library we chose is the Encog machine learning library[4.1], which developed by the Heaton research and written in the Java programming language. The Encog supports various of the machine learning algorithms, including all the techniques that was decided to use in order to build a neural network from the previous chapter [III]. Moreover, another advantage of using the Encog library is that it is designed and built to work well with the multithreading and multicore architecture, which can help to speed up the performance of the classifier.

Moreover, the Encog library also supports the data normalizing process. The numeric data which is the extracted characteristics from the writing samples that were produced by the feature extractor will need to be normalized before it can be used. The normalization process helps to improve the performance of the neural network's learning process. The data after being normalized will be fed directly to the neural network for further processing.

As decided from the previous chapter [III], we will make use of the Encog library to build the neural network with three different layers, one input layer, one output layer and one hidden layer. The neural network classifier will be constructed dynamically when the information about the training and testing data is given. The reason is because we need to know the total number of all stylometric features, which also including the flexible features, that will be used in the feature extractor component in order to initialize the number of neurons in the input layer, since the total number of neurons in the input layer is equal to the number of features that is used in the feature extractor component. For the output layer, the number of neurons will equal to the total number of persons that appears in both training and testing sets. In the hidden layer, the number of neurons will be the average number of neurons are used in both input and output layers.

The neural network classifier will receive the characteristics of the writing samples that were extracted by the feature extractor component as inputs. The received inputs will then be used to train the neural network so it can learn the differences in the writing patterns of each writer. After the neural network has been completely trained, it can be used to predict the original writer of any randomly given writing.

4.2.4 Support vector machine classifier

For building a support vector machine classifier, we also decided to choose an existing library instead of implementing one for saving time and minimize the number of possible bugs in the prototype, as we did with the neural network classifier. The support vector machine library we chose to use is Libsvm [4.2], developed by the National university of Taiwan. The Libsvm library supports the various types of the kernels and the classifying techniques such as the one against one technique for multi classifying. The Libsvm library covers everything we need to build our support vector machine classifier.

Moreover, the Libsvm library also goes with an automation script that helps to perform the cross validation on the dataset in order to find the most optimal value for the cost parameter. A small problem is that the script is written in Python programming language. Yet we were able to solve this problem by while we still allow the script to run in the Python environment, we use Java to intervene to the input and output streams of the script. In that way, we can capture the produced results from the script and transform them to the normal variables in Java for future using.

The support vector machine classifier will receive the characteristics of the writing samples that were extracted by the feature extractor component as inputs. The received inputs will be processed by the classifier through a training process, which will result in a model. The trained model will be able to distinguish different writings from different authors and can be used later to predict the original writer of any randomly given writing.

Although the support vector machine technique requires less configurations compares to the neural network technique, choosing kernel is still an important and mandatory task to do. Choosing inappropriate kernel may result in reducing the performance of the support vector machine classifier. As the discussion from the above chapter, on the one hand, the sigmoid activation function was chosen for the neural network classifier because it fits well to the feedforward approach and to the dataset, while on the other hand, the sigmoid kernel was chosen for the support vector machine classifier because it raises the similarity between these two types of classifier. A concern is raised about how this choice will affect to the performance of the support vector machine classifier in term of qualitative and quantitative attributes, maybe if there is another kernel is used for the support vector machine classifier, we might have a better result. Based on that concern, we performed a small test on the datasets that we will use in this research. We performed a test of using the support vector machine technique for the classification task, runs on the same dataset but with different kernels. This helps us to observe which kernel will work best for the support vector machine classifier with the given datasets.

Comparing the performance of kernels on diffrent corpuses

Kernel / Dataset	Sigmoid	RBF	Polynomial
B-G Dataset	60.5%	0%	14.5%
Enron Dataset	48%	51.3%	10.6%
Average	54.25%	25.65%	12.55%

Bases on the experiments between the different kernels and the given datasets, we can see that the sigmoid kernel performed well in the B-G dataset compares to the RBF and the polynomial kernels. In the Enron dataset, the RBF kernel outperformed the sigmoid and the polynomial kernels. The performance of the sigmoid kernel is a little bit behind the RBF kernel. By calculating the average on the performance of each kernel in both of the given datasets, it is clearly to see that the sigmoid kernel is an optimal option to use for the support vector machine classifier. This is also an advantage for us when we compare the performance between the neural network and the support vector machine classifiers since as discussed on the previous chapter [III], by using the same sigmoid kernel for both type of classifiers, we can raise the similarity between them, which will help to generalizing the performance result better.

4.2.5 Clusterer

In order to develop the clustering component for the prototype, we decide to take the advantage of using the existing library that supports unsupervised learning algorithms to save time and minimize the number of possible errors in the prototype later. The library is developed by Thomas Abeel [4.3] and written in the Java programming language. The library supports various type of clustering algorithms such as K-mean and SOM, which covers everything we need to build our clusterer. The developed clusterer will receive the extracted characteristics of the writing samples as inputs and produce a set of clusters as outputs, which are lists of writers that share similarities in their writing patterns. As discussed in the previous chapter [III], the algorithm that will be used in the clustering process is self-organizing map (SOM), which requires no configuration in order to perform the clustering process.

4.3 Supporting components

To be continued, the components that act as supporters in the prototype will be described. Although they are not listed as the principal components, but their existence contributes for perfecting the prototype and makes it entirely function-able.

4.3.1 Database management component

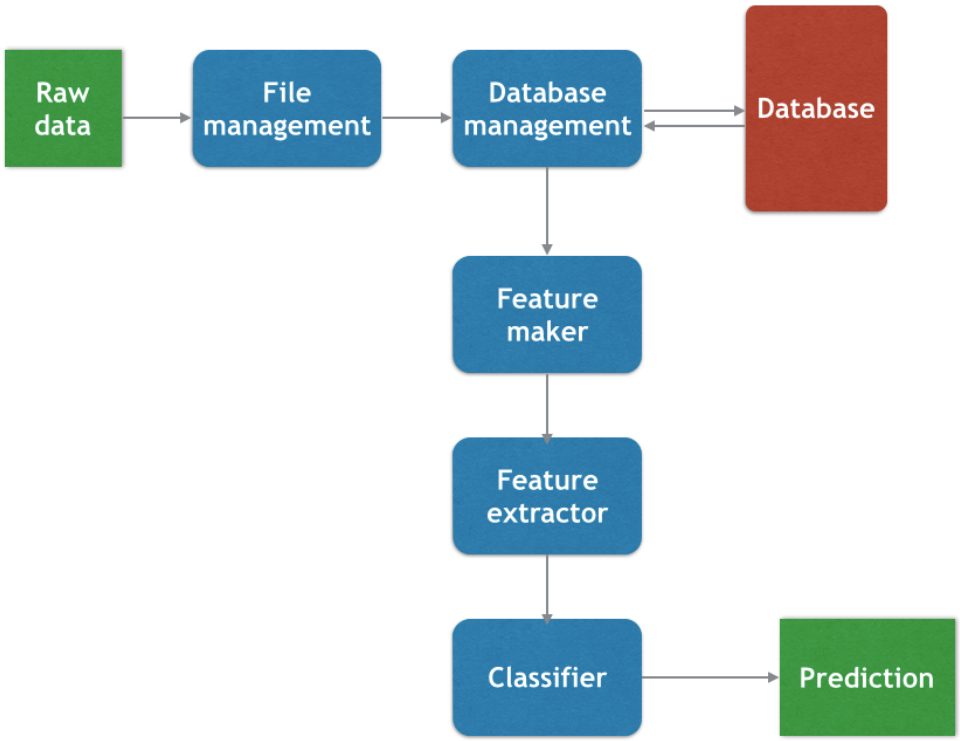
All of the data, including the writing samples and the information about the writers in both training and testing sets will be stored in a MySQL database. The main responsibility of the database management component is to provide the raw data to the other components in the prototype when it is needed. The database has two tables, which are TrainingSet and TestingSet tables. The TrainingSet table will store all the information about the writers such as their names and their writing samples, which will be used later by the prototype for the training process of the classifier component. The TestingSet table stores the testing samples and the information about the original writers of those testing samples, if available, for using later in the predicting process.

4.3.2 File management

Originally, data about the information of the writers and their writings is stored in the physical text files before they can be transferred to the database system. The file management component is responsible for reading the text files and put their contents to the database with the help from the database management component. Although this will be a time-consuming process, it only happens one time. When all of the data is stored in the database, the prototype can have faster access to any available raw data in the database at any time, than having to perform the file reading process, which is a low-performance task, every time it needs the data to work.

4.4 Data processing workflow in the prototype

The raw data, with are the text files, will be read by the file management component. After that, with the support of the database management component, the contents that are read from the text files will be put to the database for storing. The authorship identification process starts when the feature maker accesses to the database for making the new features. The new features will be added to the feature extractor component. The feature extractor component will get the data in the database via the database management component and extract the characteristics of the retrieved data. Then the extracted characteristics will be transferred to the classifier component, which is responsible for creating and training a predicting model, that will be used later for predicting the original writers of the given testing samples in the database.



Demonstration for data processing steps in the prototype.

V. Evaluation

Next, various experiments will be performed to examine the qualitative (the accuracy level) and the quantitative (the performance time) attributes of the prototype in order to compare its original performance abilities to the new performance abilities when the Alpha, Beta, Gamma and Delta improvement projects are applied to the initial solution. The results will be marked up for improving the readability of the reader. The raw results can be found in the appendix section at the end of this paper.

5.1 Experimental setup

In order to keep the consistency in the results of the performed tests, all of the tests will be performed on one single machine only. The configuration of the machine is as following.

Components' information

ID	Component	Info
01	CPU	2.4 GHz Intel Core 2 Duo
02	RAM	08 GB 1067 MHz DDR3
03	OS	OS X 10.9.2
04	Java runtime	1.7.0_15
05	Java Development	07
06	Netbean IDE	7.4
07	MySQL	5.5.29

When the tests are being performed, besides the default services and programs started by the operating system, only Netbeans IDE and MySQL applications are started under the user privilege.

5.2 Dataset

There will be two different kinds of the dataset that are used when evaluating the qualitative and quantitative attributes of the prototype. The first dataset will be used as a training dataset. This dataset will contain the sample writings from particular writers and will be used to train the created model of the classifier. The second dataset will be used for the testing purpose. The data in the training dataset and testing dataset is different.

In this evaluation, two different corpuses will be used in order to guarantee the neutral property in the produced results by the prototype. Using different corpuses helps to avoid the prototype may have a high performance result if the data in the corpus is compromised or in the opposite case. The first corpus is called the Brennan-Greenstadt Adversarial Corpus [5.1]. This corpus contains the writings of 46 different users, and each user has in total minimum of 6500 words, which is spread over between 10 to 20 different writings. For each user, three random writings will be separated from the corpus to use for the testing purpose. The remaining writings will be put to the training dataset. The second corpus is used in this research comes from the Enron company, which contains the data as emails from 50 different users, mostly the senior managements of Enron. This dataset was originally made public by the Federal Energy Regulatory Commission during its investigation about the bankrupt of Enron company [5.2]. Each user in Enron dataset has around between 10 to 12 different writings. The division for the training set and testing set of the Enron corpus will be done as in the first corpus, three random writings will be separated into a testing dataset, and the remaining will be kept and used as the data in the training dataset.

In general, in total there will be 96 different writers are involved in this evaluation. The total number of the training samples is 1170 items. The total number of the testing samples is 386 items. These datasets will be used to test against the qualitative and the quantitative attributes of the prototype.

There is one additional corpus, as an extra request from the Kalooga company, which is predicting who is the real author of the Bit-coin system by analyzing the Bit-coin's white-paper [5.3]. As a brief description, Bit-coin is created as an online cash value trading system by an anonymous person. Later on, a document named Bit-coin white-paper was published to give to the Bit-coin's user a better understanding about the Bit-coin system. The author of the Bit-coin white-paper named himself as Satoshi Nakamoto, which is not a real person. According to N. Hajdarbegovic [5.4], a group of linguistic researchers names Nick Szabo as the author of the Bit-coin white-paper after they analyzed writings from a group of people that they suspected who can be the real creator of the Bit-coin system. The suspicious list contains the following persons: Dorian S Nakamoto; Vili Lehdonvirta; Michael Clea; Shinichi Mochizuki; Gavin Andresen; Nick Szabo; Jed McCaleb; Dustin D Trammel; Hal Finney; Wei Dai; Neal King, Vladimir Oksman and Charles Bry. In this research, 05 authors was chosen, which are Dustin, Finney, Nick, Shinichi and Weidai because they have various of online publics and can be accessed easily. Their publics will be used as training samples for the prototype in order to find their writing patterns. Each author will be presented by five different training samples. The prototype will try to predict within the five of those authors, who actually wrote the Bit-coin white-paper.

5.3 Metrics

5.3.1 Metrics for measuring the qualitative and quantitative values

For evaluating the qualitative attribute (the accuracy level) of the implemented prototype, various test cases will be performed with randomly chosen writers in the training set in order to determine the true positive and the false negative rates from the results. The prototype will be firstly trained with training samples of chosen writers. Then it will predict the original writers of testing samples, from that we will be able to compare if the prototype's predictions match to the original writers of the testing samples. For this given test, the chosen writers that will appear in the training set will be also appeared in the testing set. If the prediction from the prototype matches to the given information of the writer, this will contribute to the true positive rate of the prototype. Otherwise, it will contribute to the false negative rate, which means the prototype incorrectly recognizes the original writers of the testing samples. The true positive value is also the recall value of the prototype. The higher the recall value is, the better the accuracy level that the prototype has. In this given test, for the convenience, the original information of the writers in the testing samples will be given to the prototype so that we can access to the results faster, by comparing and giving true/false values only. Below is the access table for comparing the predicted value from the prototype with the actually given value.

Confusion table for classification problem

Comparison	Predicted value	Predicted value
Given original value	Same-> True positive	Different->False negative

For evaluating the quantitative attribute (the performance ability) of the prototype, different tests will be performed with different amount of randomly chosen writers in the training set. The execution time of the prototype will be recorded and used later for the comparison in the change in the executing speed of the prototype when different type of classifier are used.

5.3.2 Bit-coin whitepaper test

The last evaluation, as a request from the Kalooga company, will be about predicting the original author of the Bit-coin white-paper, as mentioned before in the above Dataset section. The result from this experiment will be presented later in the upcoming section.

5.4 Running

5.4.1 Qualitative and quantitative tests on the initial solution

The initial solution prototype, which represents for the existing solution from the research works of the other researchers in the authorship identification research area, will be tested with various number of persons in the training dataset. This process helps to indicate the performance of the prototype, to observe how the performance changes when the difficult level of the test is increased, by increasing the number of different persons in the dataset. Persons that will appear in the training and testing datasets will be picked up randomly by the prototype. The number of persons that will be tested is increased gradually from low to high, beginning with two different persons in the dataset and ending up to thirty different persons in the dataset. When the different amount of persons in the dataset is tested, each test will be run repeatedly in three times, each time with a different list of persons and different types of classifier. The qualitative attribute of the prototype will be presented by the percentage of the number of testing samples that are correctly identified for their original writers, over the total number of testing samples. The quantitative attribute will be measured by the execution time of the prototype in order to completely process the dataset, from the data normalizing step till the data prediction step. The testing procedure can be seen as below, and this procedure will be applied the same to each data corpus. A specific note here is that for the support vector machine classifier, each independent test will be performed one time only, because the result from the support vector machine classifier is consistency for the same training and testing datasets no matter how many times the test is performed. On the other hand, the neural network classifier tends to give different result because of its natural random attribute when calibrating the network during training process, as explained in previous chapter [IV], thus for each independent test with the neural network classifier, it will be performed three times on the same datasets. The measured qualitative and quantitative values will be the average value of these three performed tests.

The procedure for evaluating the performance

ID	No. of persons	No. of independent test times	Classifier	Number of the corpuses
01	2	3	NN and SVM	2
02	3	3	NN and SVM	2
03	5	3	NN and SVM	2
04	10	3	NN and SVM	2
05	30	3	NN and SVM	2

In total, the number of the performed tests with the support vector machine classifier is 30 tests, while the number of the performed tests with the neural network classifier is 90 tests.

5.4.2 Bit-coin whitepaper test

In this experiment, both of the neural network and the support vector machine classifiers will perform the classifying test on the same dataset. The objective of this test will be identifying the original author of the given Bitcoin white-paper, bases on the provided list of the suspected writers and their writings. In this test, the neural network classifier will be also run three times, with the same reason from the previous section, which is because of the instability in the neural network.

5.4.3 Performance experiments on proposed improvements

Up to this point, the classifier that outperformed the other one in the performance tests will be used in the next experiments on testing the improvement projects for the prototype. This decision helps to reduce the spending time for the evaluation section while still guarantees to deliver the best result. The qualitative and quantitative attributes of the prototype will be re-measured whenever an improvement project is applied. Besides the project Alpha, which only focuses on improving the qualitative value of the prototype, thus the testing procedure for the project alpha will be the same as the testing procedure that was applied to the initial solution. For the beta, gamma and delta improvement projects, for each project, the prototype will be tested three different times on each corpus, each test will be performed with a different list of randomly chosen persons in the dataset in order to guarantee the impartiality in the testing results from each corpus.

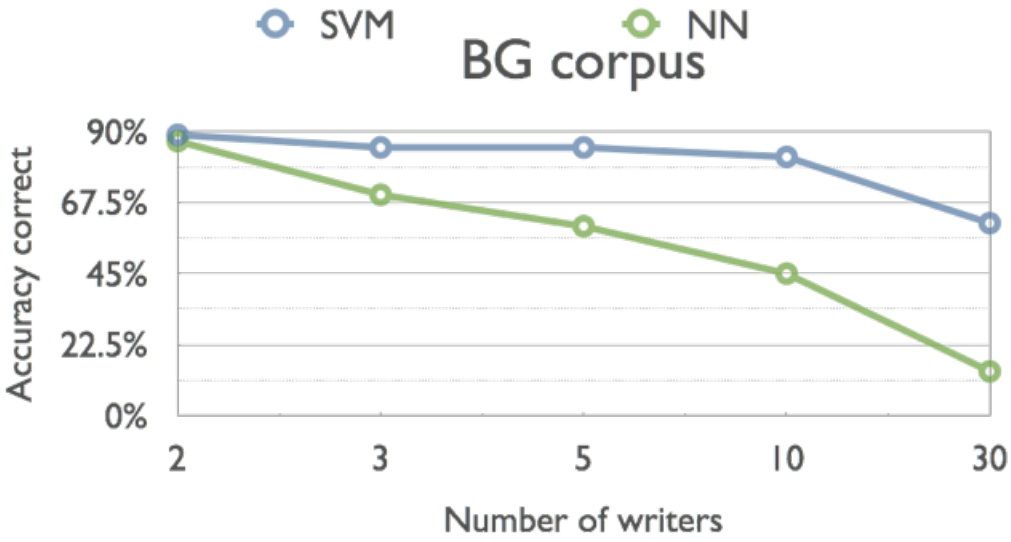
5.5 Result for qualitative and quantitative tests

The raw results from the evaluation can be found in the appendix section. For the presentation purpose, all of the results will be marked up in order to improve the understandability. The tests were performed with different number of writers in the training and testing sets.

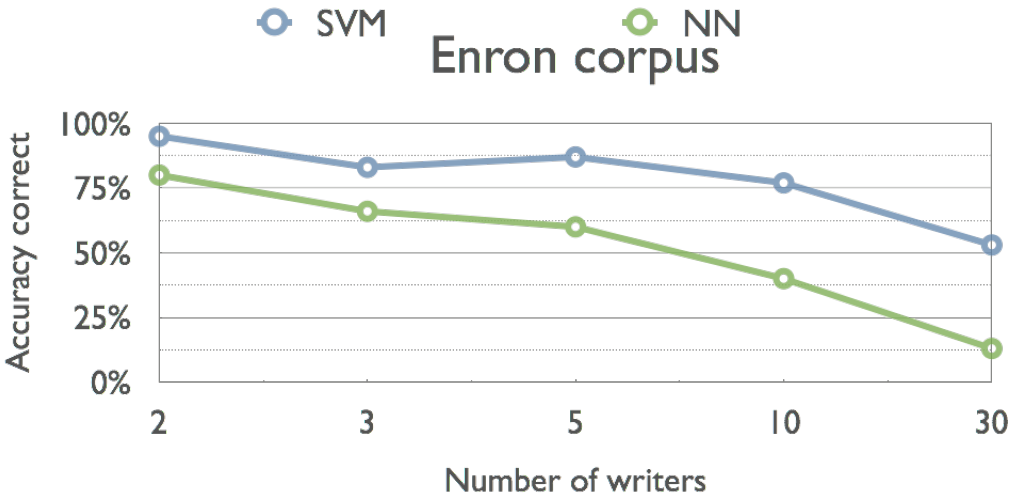
5.5.1 Qualitative and quantitative test on the initial solution

In this section, the qualitative and quantitative tests will be performed on the initial prototype, which is built from the existing research works from other researchers in the authorship identification research area. The results from these experiments will be set as the standard points for the further comparisons when the improvement projects for the prototype are applied. The raw results are presented in Appendix A.1 and A.2 sections.

a. Prototype's qualitative performance of the Support vector machine and neural network classifiers with Brennan-Greenstadt Adversarial Corpus and Enron Corpus



The comparison on the accuracy level of the SVM and NN classifiers on the B-G corpus



The comparison on the accuracy level of the SVM and NN classifiers on the Enron corpus

Looking at the results from the qualitative tests on the prototype when different classifiers are used, it is easy to see that on both corpuses, the classifier that built on the support vector machine technique gained a better result in the accuracy level compares to the classifier that built on the neural network technique. It is notable that during the authorship identification task, when the number of different persons in the dataset that the prototype must distinguish is increased, the accuracy level is decreased. In the experiment, when the number of persons in the dataset is increased from 02 to 30 persons, the classifier with the support vector machine technique lost 35% of the accuracy level in the authorship identifying task while, with the classifier with the neural network technique, the loss in the accuracy level is up to 70%. More details on the presented graphs can be found in the below tables.

The accuracy level of the SVM and NN classifiers on B-G corpus

Number of writers	SVM Accuracy	NN Accuracy	Corpus	No. of tests	SVM Dataset	NN Dataset
02	89%	87%	BG	03	023/006	027/006
03	85%	70%	BG	03	038/009	038/009
05	85%	60%	BG	03	065/015	067/015
10	82%	45%	BG	03	126/030	120/030
30	61%	14%	BG	03	371/89	360/090

*Note: In SVM Dataset and NN Dataset columns, ddd/ddd means number of samples in the training dataset / number of samples in the testing dataset.

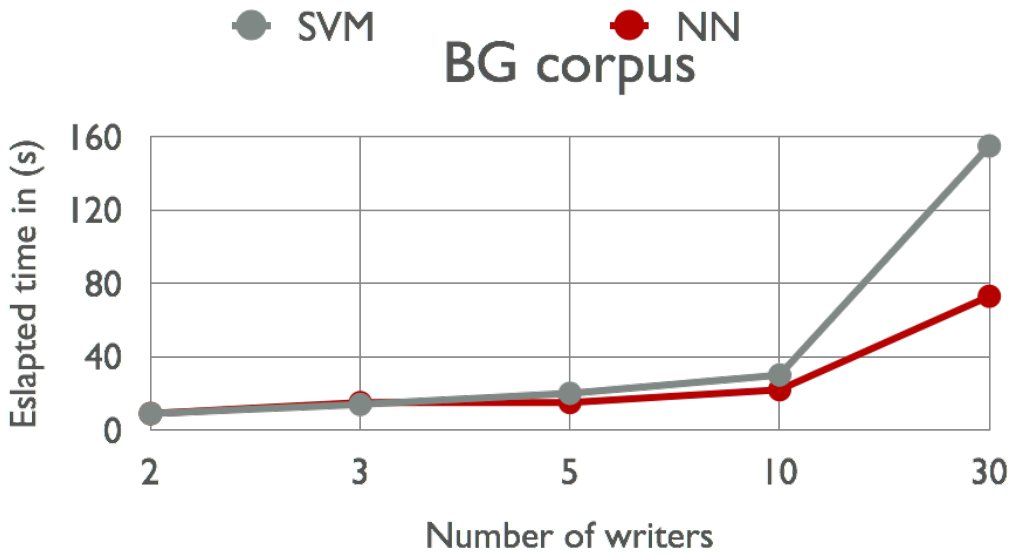
The accuracy level of the SVM and NN classifiers on Enron corpus

Number of writers	SVM Accuracy	NN Accuracy	Corpus	No. of tests	SVM Dataset	NN Dataset
02	95%	80%	Enron	03	020/006	020/006
03	83%	66%	Enron	03	030/009	030/009
05	87%	60%	Enron	03	050/015	050/015
10	77%	40%	Enron	03	100/030	100/030
30	53%	13%	Enron	03	300/090	300/090

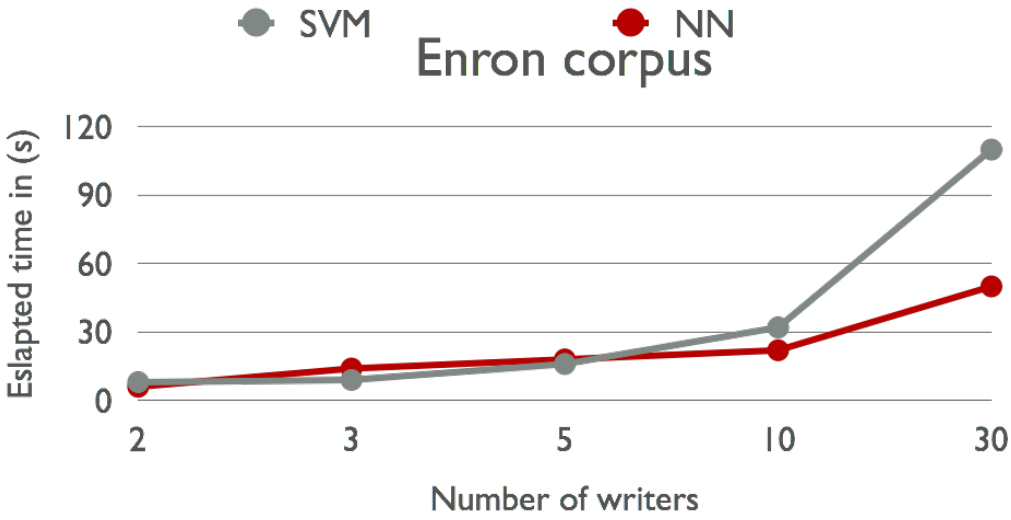
*Note: In SVM Dataset and NN Dataset columns, ddd/ddd means number of samples in the training dataset / number of samples in the testing dataset.

In order to improve the readability, the presented value in the above tables is an average number of the combined values from the different performed tests. Detail results can be found in the Appendix section.

a. Prototype's quantitative performance of the Support vector machine and Neural network classifiers with Brennan-Greenstadt Adversarial Corpus and Enron Corpus



The comparison on execution time of the SVM and NN classifiers on the B-G corpus



The comparison on execution time of the SVM and NN classifiers on the Enron corpus

In the quantitative tests on the prototype, when the number of different persons in the dataset is below ten persons, there is no significant different in the execution time of both support vector machine and neural network classifiers. When the number of persons in the dataset is increased to 30 different persons, there is a dramatic change in the execution time of the classifier with the support vector machine technique compares to the classifier with the neural network technique. When the number of persons in the dataset is increased from 10 persons to 30 persons, the execution time of the support vector machine classifier is increased by four times, compares to two and a half times of increment in the execution time of the neural network classifier. More details on the presented graph can be found in below tables.

The performance of the SVM and NN classifiers on the B-G corpus

Number of writers	SVM Run time	NN Run time	Corpus	No. of tests	SVM Dataset	NN Dataset
02	009	002-009	BG	03	023/006	027/006
03	014	004-015	BG	03	038/009	038/009
05	020	007-015	BG	03	065/015	067/015
10	030	011-022	BG	03	126/030	120/030
30	155	048-073	BG	03	371/89	360/090

*Note: In SVM Dataset and NN Dataset columns, ddd/ddd means number of samples in the training dataset / number of samples in the testing dataset.
**Note: In SVM RunTime and NN RunTime columns, ddd/ddd means the range of the execution time of the neural network classifier.

The performance of the SVM and NN classifiers on Enron corpus

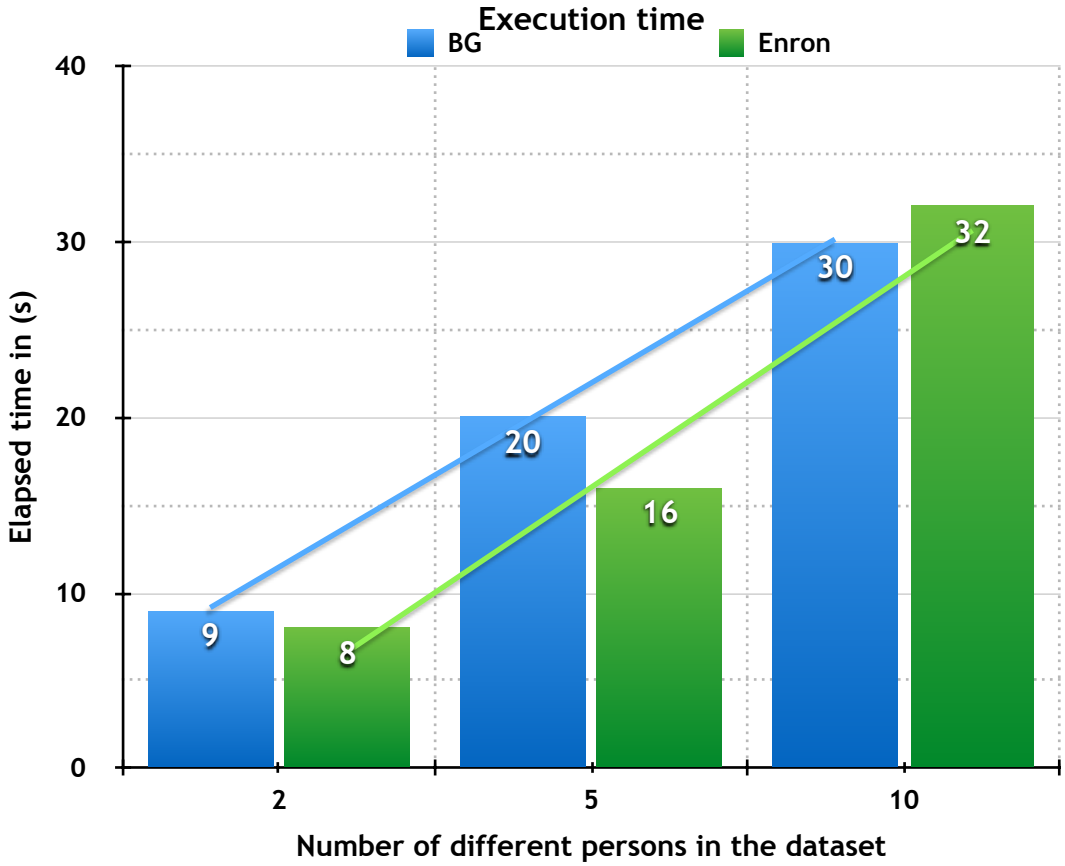
Number of writers	SVM Run time	NN Run Time	Corpus	No. of tests	SVM Dataset	NN Dataset
02	008	003-006	Enron	03	020/006	020/006
03	009	004-014	Enron	03	030/009	030/009
05	016	005-018	Enron	03	050/015	050/015
10	032	010-022	Enron	03	100/030	100/030
30	110	050-280	Enron	03	300/090	300/090

*Note: In SVM Dataset and NN Dataset columns, ddd/ddd means number of samples in the training dataset / number of samples in the testing dataset.
**Note: In SVM RunTime and NN RunTime columns, ddd/ddd means the range of the execution time of the neural network classifier.

In order to improve the readability, the presented value in the above tables is an average number of the combined values from the different performed tests. The detail result can be found in the appendix section.

Since the classifier with the support vector machine technique outperformed the classifier with the neuron network technique in the qualitative comparison, later experiments will use the support vector machine classifier as the default classifier, this helps to save time and spend more focus on improving the parts that already better than the other alternative parts which have less impressions (The neural network classifier in this case).

There is an interesting point about the relationship between the number of different persons in the dataset and the execution time. When the size of the dataset is increased by double, it seems that the prototype also needs a twice amount of time to finish the task. This can be an excellent performance measuring point when comparing the performance of the improvement projects when they are applied to the prototype later.



5.5.2 Bit-coin whitepaper test

In this Bit-coin white-paper experiment, the prototype will try to predict the original author of the Bit-coin white-paper in the given list of authors' names that have writing samples in the dataset. According to Nermin Hajdarbegovic [5.4], he claimed that linguistic researchers named Nick Szabo as the author of the Bit-coin white-paper.

a. With the Support vector machine classifier

The result of the Bit-coin whitepaper test from SVM classifier

Number of writers	Train/Test samples	Original	Result
06	030/001	Bit-coin	Nick

*Note: In the Train/Test samples column, ddd/ddd means number of samples in the training dataset / number of samples in the testing dataset.

b. With the Neural network classifier

The result of the Bit-coin whitepaper test from NN classifier

Number of writers	Train/Test samples	Original	Result
06	030/001	Bit-coin	Finney, Weidai

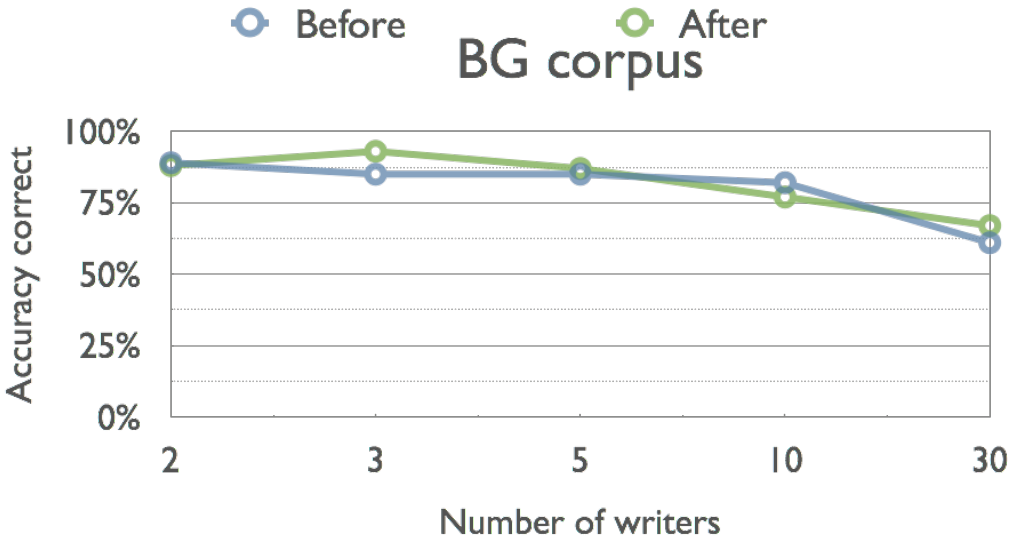
*Note: In the Train/Test samples column, ddd/ddd means number of samples in the training dataset / number of samples in the testing dataset.

By using the combined set of stylometric features from the existing works of other researchers in the authorship identification research area, the prototype predicted that the original author of the Bit-coin white-paper is Nick Szabo when the support vector machine classifier is used. This result is similar to the information provided by Nermin Hajdarbegovic in his article. On the other hand, when the neural network classifier is used, with the same stylometric feature set, the prototype predicted that the original author of the Bit-coin whitepaper should be Finney and Weidai.

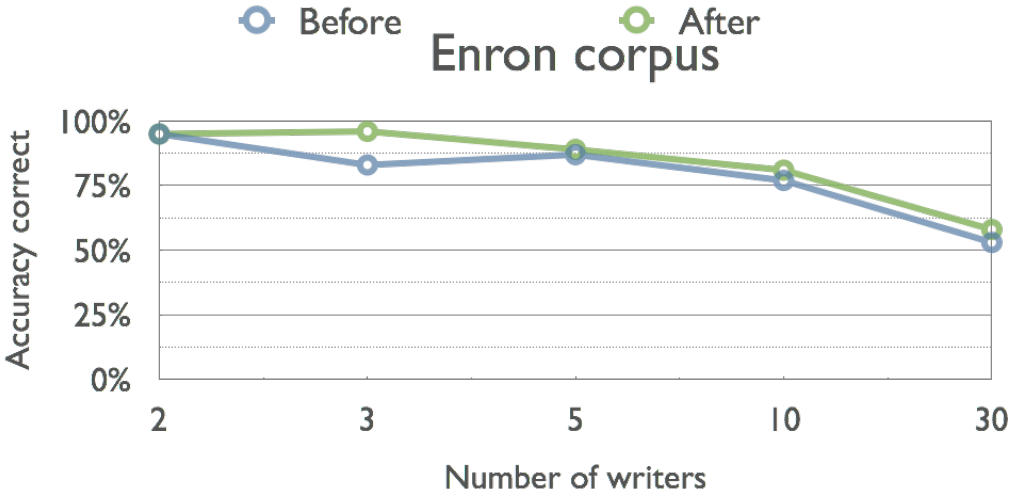
5.5.3 Project Alpha

In the project Alpha, we focused on improving the qualitative attribute of the initial prototype by adding more features to the feature extractor component in order to help the extractor can get more hidden patterns in the writing style of a particular person. More information about this project can be found in previous chapter [III]. The raw results are presented in the Appendix A.3 sections.

a. Prototype's qualitative performance before and after the project Alpha with Brennan-Greenstadt Adversarial and Enron corpuses



The accuracy level of the prototype on the BG corpus before and after applying the project Alpha



The accuracy level of the prototype on the Enron corpus before and after applying the project Alpha

From the collected result, in general, the accuracy level in the authorship identification task of the prototype slightly improved by 02% to 5% after the project Alpha is applied.

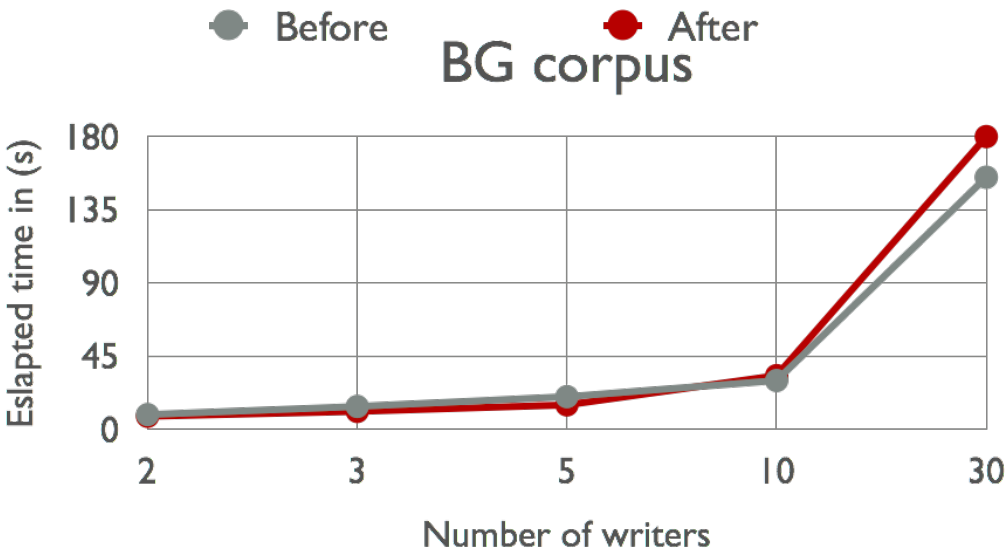
We also performed an experiment in finding the original author of Bit-coin white-paper again after the project Alpha is applied to the prototype. Below is the predicted results from the prototype when both classifiers is used.

Result of the Bit-coin whitepaper test from SVM and NN classifier

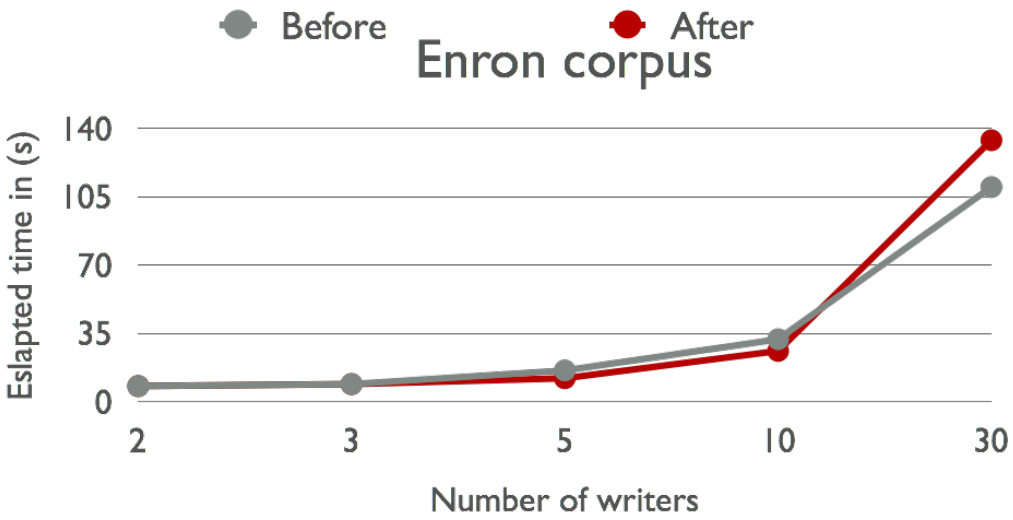
Number of writers	Train/Test samples	Original	SVM Result	NN Result
06	030/001	Bit-coin	Weidai	Dustin, Finney

*Note: In the Train/Test samples column, ddd/ddd means number of samples in the training dataset / number of samples in the testing dataset.

b. Prototype's quantitative performance before and after the project Alpha with Brennan-Greenstadt Adversarial and Enron corpuses



The execution time of the prototype on the BG corpus before and after applying the project Alpha



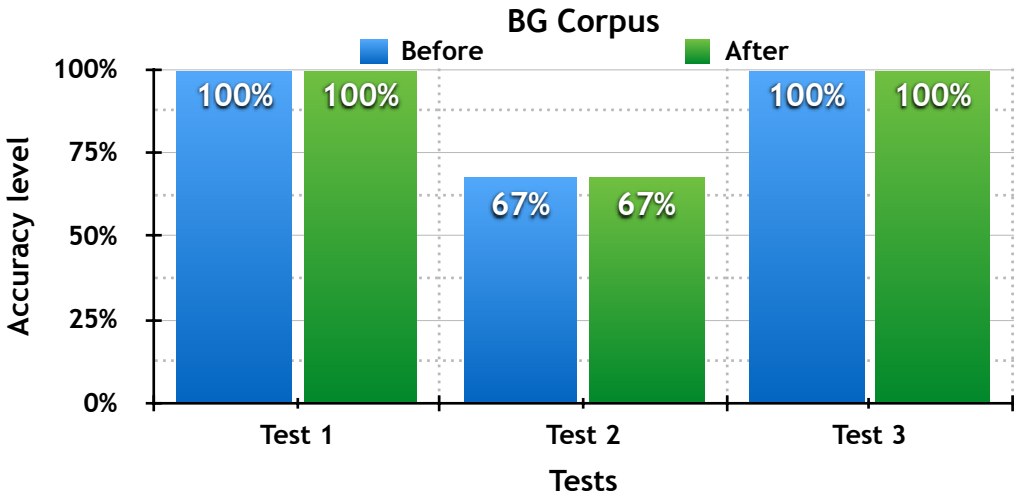
The execution time of the prototype on the Enron corpus before and after applying the project Alpha

It is easy to see that the performance of the prototype is slightly decreased. This is reasonable because the training process of the classifier is slowed down because after the project Alpha is applied to the prototype, more collected data from the feature extractor is provided to the classifier, which results in an increment in the training time for the created model.

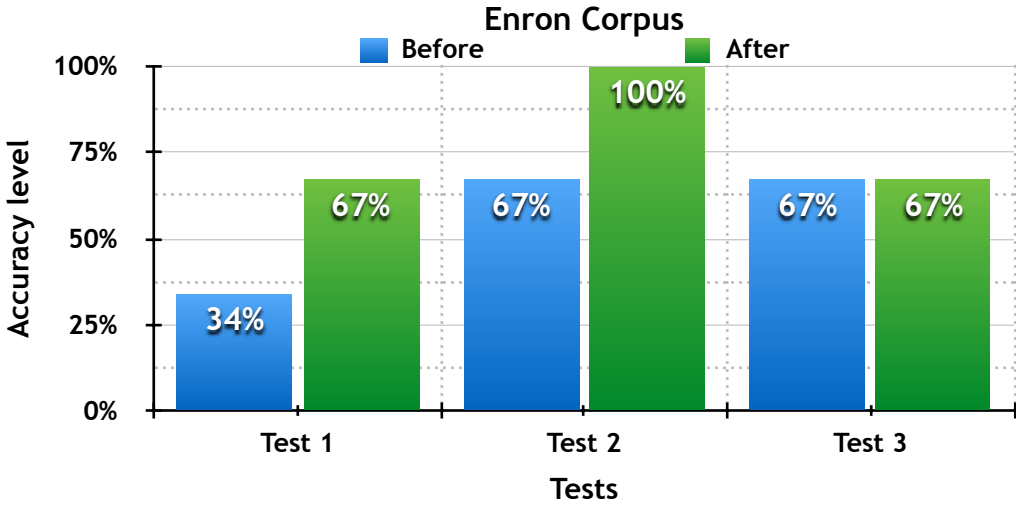
5.5.4 Project Beta

In the project Beta, it aims to improve the qualitative and the quantitative attributes of the initial solution by making use of the unsupervised learning techniques, which aims to decrease the dataset's size. More information about this project can be found in previous chapter [III]. The raw results are presented in the Appendix A.4 sections.

a. Prototype's qualitative performance before and after the project Beta with Brennan-Greenstadt Adversarial and Enron corpuses



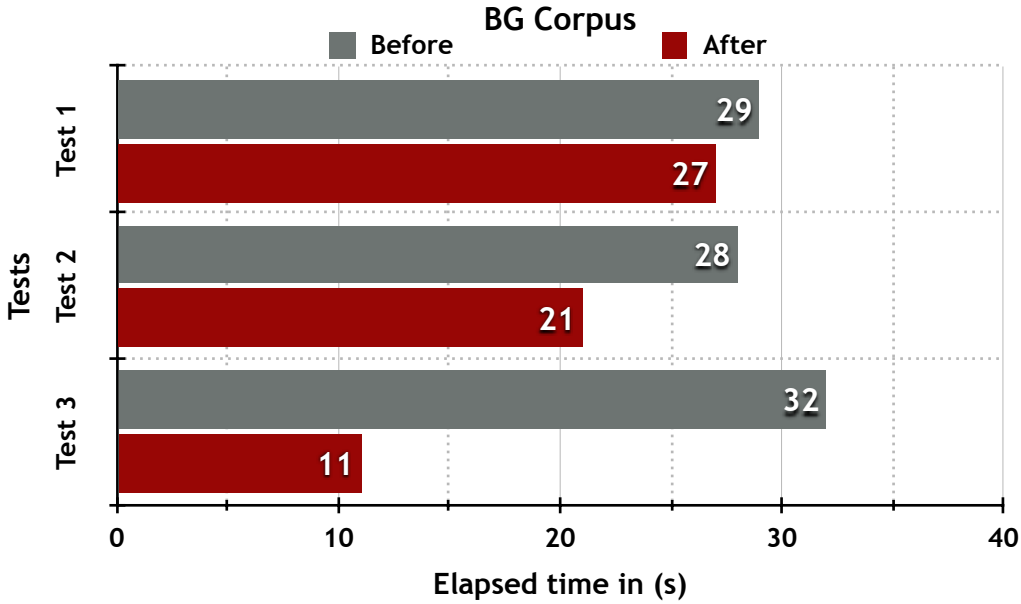
The accuracy level of the prototype on the BG corpus before and after applying the project Beta



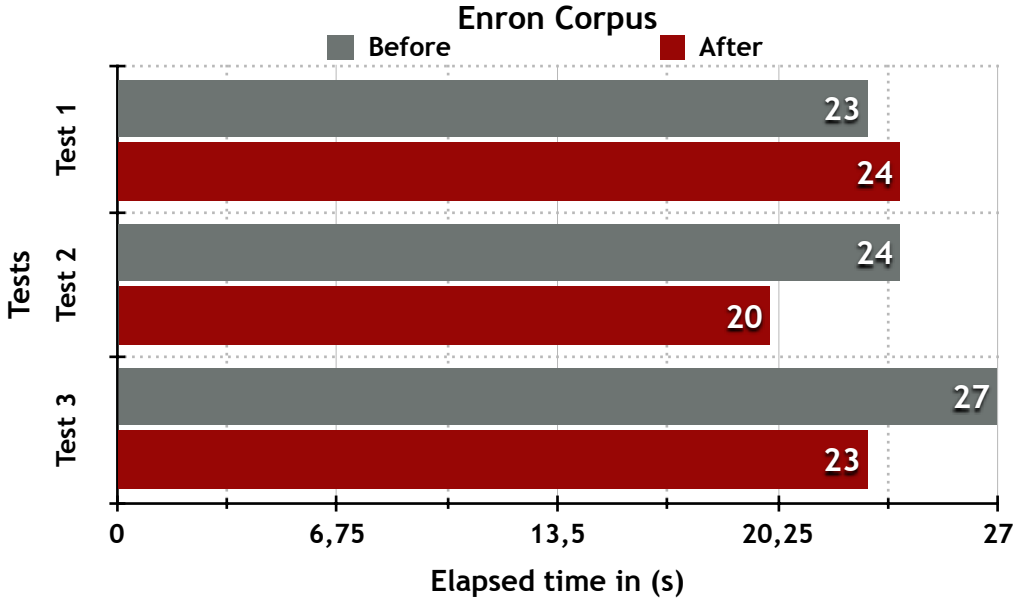
The accuracy level of the prototype on the Enron corpus before and after applying the project Beta

When the qualitative experiment is performed on the BG corpus, the accuracy level stays neutrally. In the Enron corpus, in overall, the accuracy level is increased by approximately 23%. This can be a result of the unsupervised learning technique has successfully in decreasing the number of different persons in the dataset, which helps to simplify the trained model that created by the support vector machine classifier.

b. Prototype's quantitative performance before and after the project Beta with Brennan-Greenstadt Adversarial and Enron corpuses



The execution time of the prototype on the BG corpus before and after applying the project Beta



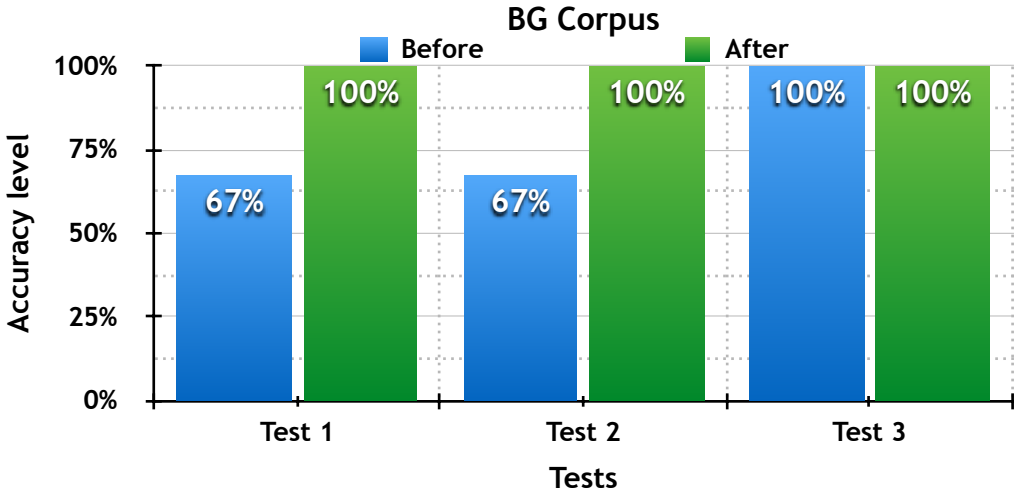
The execution time of the prototype on the Enron corpus before and after applying project Beta

The performance of the prototype has a notable improvement. In both corpuses, the execution time of the prototype after applying the project Beta is lower than the execution time of the initial prototype. This can be reasoned as the unsupervised learning technique helps to reduce the training time of the classifier by giving it a smaller dataset, while it still contains highly relevant data.

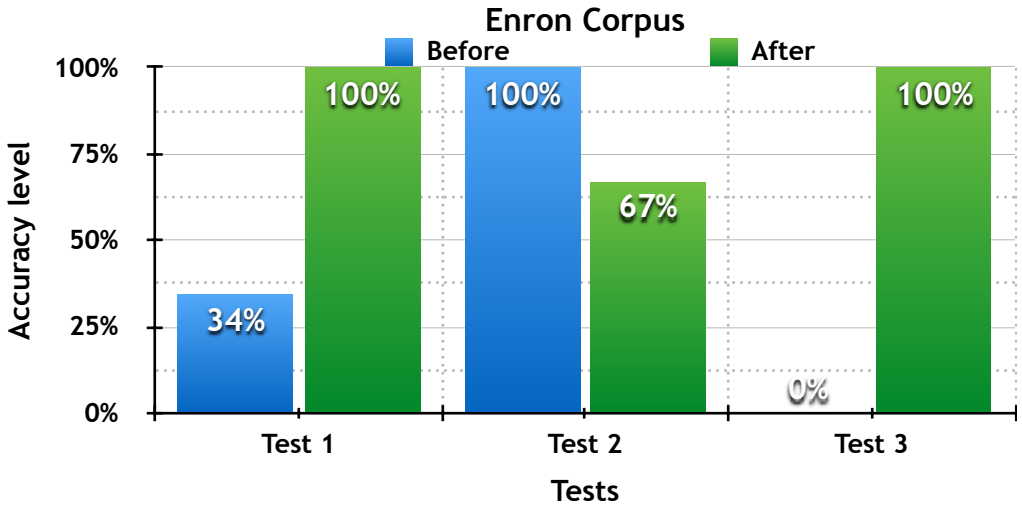
5.5.5 Project Gamma

In the project Gamma, we focused on a solution that may help the initial prototype to overcome its weakness, which is the non-linear relationship between the number of different persons in the dataset and the results of the prototype's qualitative and quantitative values. Since as the previous experiments, it is easy to see that with a higher number of different persons the prototype needs to distinguish, the lower value in the qualitative and quantitative tests it will have. We try to overcome this problem by dividing the original dataset into the smaller ones in a random way to help reducing the total training time. More information about this project can be found in previous chapter [III]. The raw results are presented in the Appendix A.5 sections.

a. Prototype's qualitative performance before and after the project Gamma with Brennan-Greenstadt Adversarial and Enron corpuses



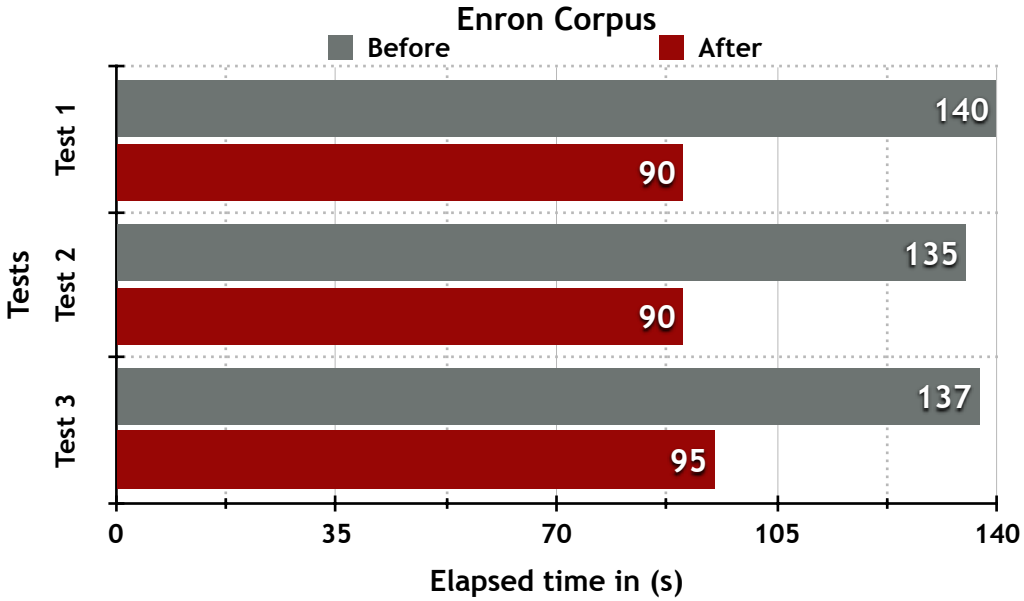
The accuracy level of the prototype on the BG corpus before and after applying the project Gamma



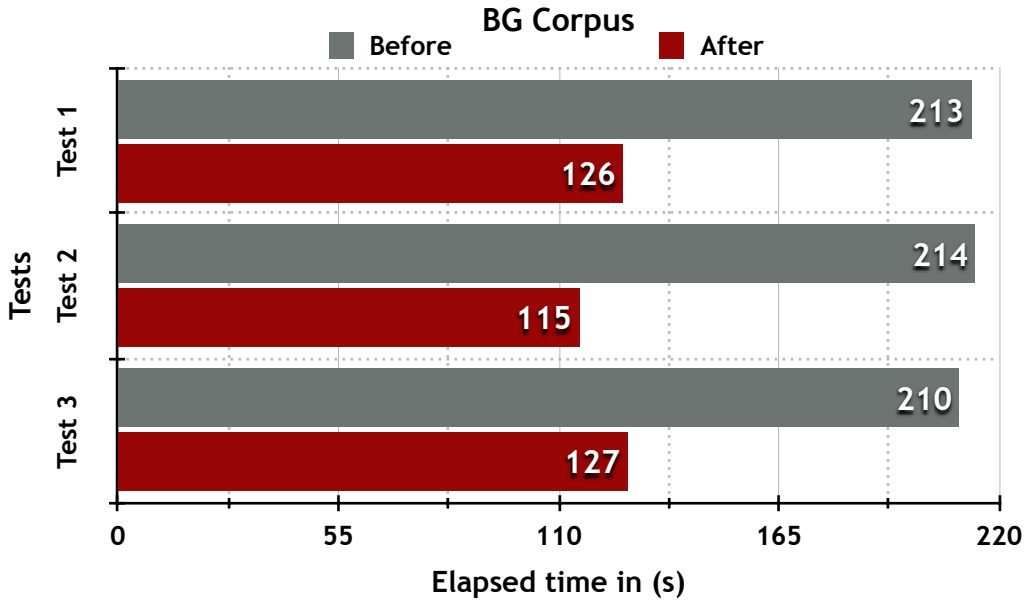
The accuracy level of the prototype on the Enron corpus before and after applying the project Gamma

After the project Gamma, the accuracy level of the prototype has a notable improvement. The performed qualitative experiment shows that the accuracy level of the prototype is increased up to 23% with the BG corpus and 44% with the Enron corpus after the project Gamma is applied.

b. Prototype's quantitative performance before and after the project Gamma with Brennan-Greenstadt Adversarial and Enron corpuses



The execution time of the prototype on the BG corpus before and after applying the project Gamma



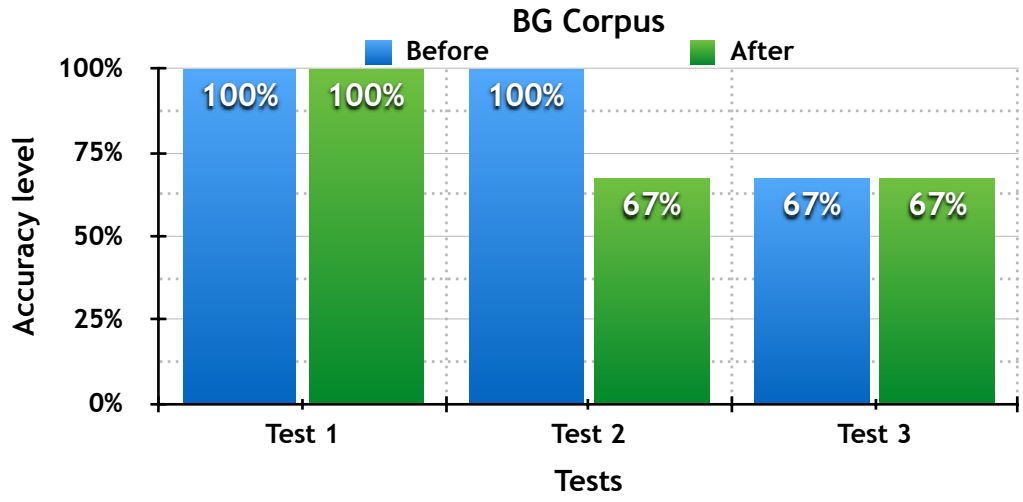
The execution time of the prototype on the Enron corpus before and after applying the project Gamma

The performance of the prototype is improved significantly after the project Gamma is applied. When the quantitative experiments were performed on both BG and Enron corpuses, the execution time of the prototype after applying the project Gamma is reduced by almost a haft of the execution time of the initial prototype.

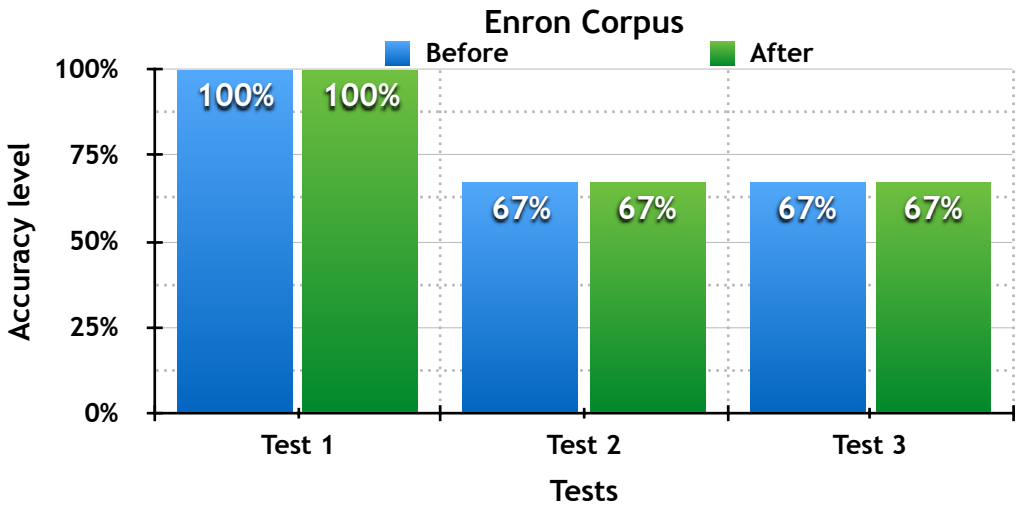
5.5.6 Project Delta

The project Delta focused on improving the quantitative attribute of the prototype by using the unsupervised learning technique in order to decrease the processing time of the authorship identification task. This is done by giving a smaller dataset but still contains highly relevant data to the classifier in order to reduce the training time for the created model by the classifier. The raw results are presented in the Appendix A.6 sections.

a. Prototype's qualitative performance before and after the project Delta with Brennan-Greenstadt Adversarial and Enron corpuses



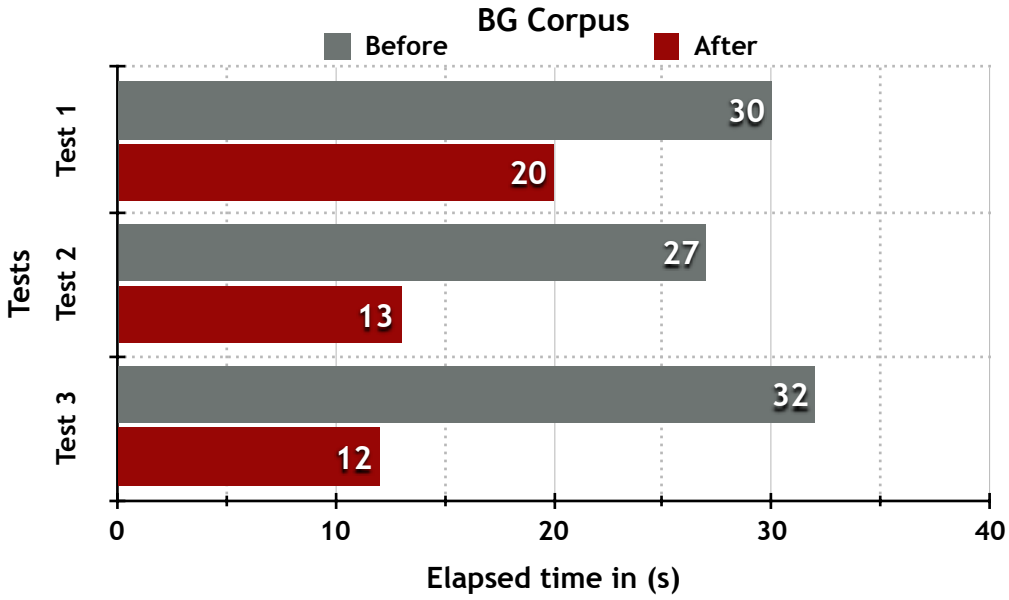
The accuracy level of the prototype on the BG corpus before and after applying the project Delta



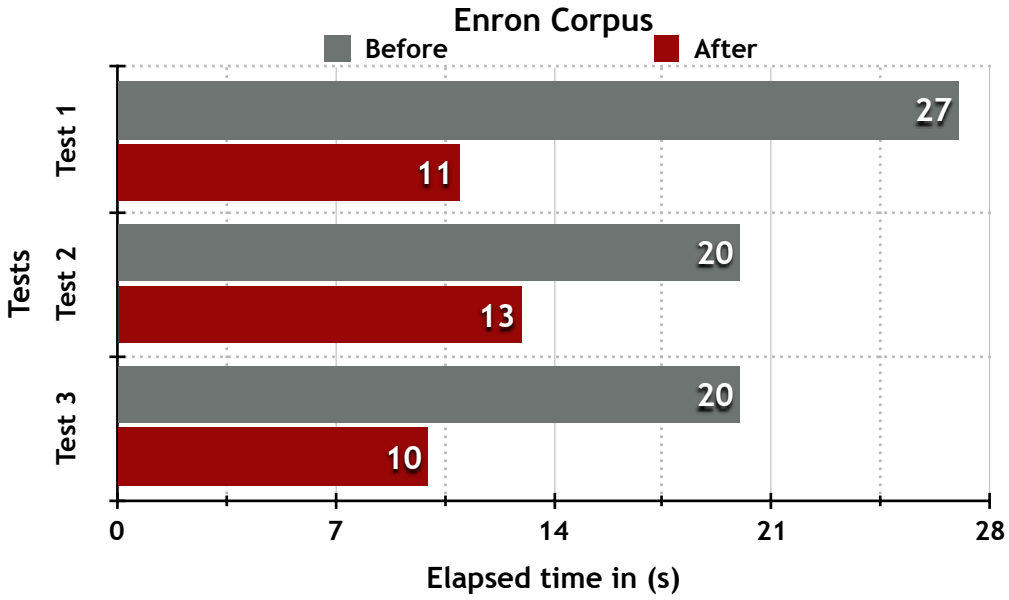
The accuracy level of the prototype on the Enron corpus before and after applying the project Delta

For the qualitative experiments on the prototype before and after the project Delta is applied, there is not much improvement on the accuracy level. With the experiment on the BG corpus, there is only a slight improvement in the accuracy level, while, with the Enron corpus, the accuracy level stays the same, with or without the presence of the project Delta.

b. Prototype's quantitative performance before and after the project Delta with Brennan-Greenstadt Adversarial and Enron corpuses



The execution time of the prototype on the BG corpus before and after applying the project Delta



The execution time of the prototype on the Enron corpus before and after applying the project Delta

The performance of the prototype is improved notably after the project Delta is applied. The performed quantitative experiments gained almost three times better result with the BG corpus and nearly four times better on the result with the Enron corpus when the project Delta is applied to the prototype.

5.6 Result interpretation

5.6.1 Initial prototype

Bases on the performed experiments on the initial prototype, which were built to represent the existing solutions from the other researcher for solving the authorship identification problem, the first impression can be drawn from the result is that there is a non-linear relationship between the number of different persons in the dataset and the prototype's accuracy level and its execution time in the authorship identification task. The higher number of different persons in the datasets, the lower value of the accuracy level and the longer execution time for the authorship identification task will be. There are two root causes for this fact.

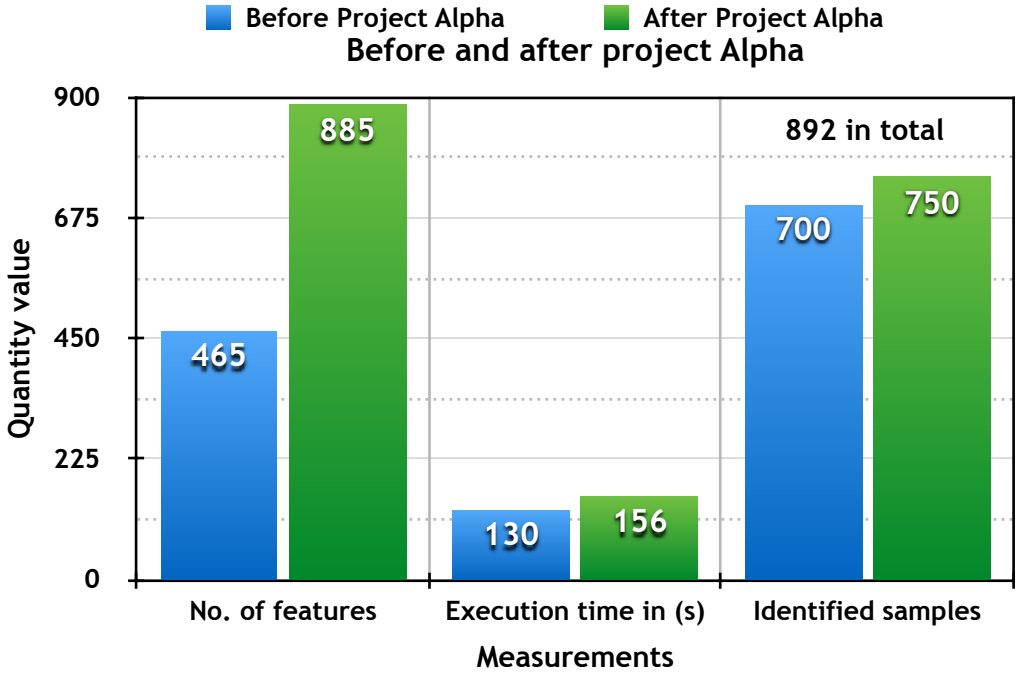
The first root cause is related to the ability of the feature extractor component in the prototype, which is powered by a set of stylometric features in finding writing style of any particular person by using their writing samples. The better the feature extractor can perform the writing style seeking task on different writing samples of a person, the more accurate the classifier can distinguish that person from other persons later during the authorship identification process. It is easy to see that the stylometric feature set which is used in the feature extractor component plays an important role in getting higher accuracy level in the authorship identification task later. When the increment in number of different persons in the dataset results in the decrement of the accuracy level, besides the excuse of different persons may share similarities in their writing styles, it also means that the stylometric feature set may be not good enough in seeking for more hidden writing patterns in the given writing samples of a person. On another hand, the higher number of patterns that the feature extractors can recognize, the longer processing time the classifier will need in order to train its predicting model, which leads to the increment in the execution time of the prototype.

The other root cause of the non-linear relationship between the number of different persons in the datasets and the prototype's accuracy level and its execution time in the authorship identification task belongs to the classifying ability of the classifier. From the result, it can be seen easily that different classifying technique can result in different produced accuracy level. From the performed experiments on the prototype when both neural network and support vector machine classifiers are used, the result can confirm the fact that the support vector machine classifier can perform better in the classification task than the neural network classifier. This is also suggested from the works of the other researchers.

In short, besides the impacting factors such as the available number of writing samples for each person in the training set, or the abilities of the stylometric feature set and the classifier on the qualitative and quantitative results of the prototype, the performance of the prototype is also highly depended on the amount of different persons in the dataset. Bases on performed experiments, a low number of different persons in the dataset usually results in the higher accuracy level and the shorter execution time.

Observing that, the improvement projects for the existing solution were proposed and applied to the initial prototype in order to sanitate the qualitative and the quantitative attributes of the prototype by focusing on providing a better dataset to the classifier. More details about the improvement projects can be found in previous chapter [III].

5.6.2 Project Alpha



Project Alpha focuses on improving the stylometric feature set in the feature extractor component with the expectation that the prototype can gain a better qualitative result. After the project Alpha has been applied to the initial prototype, the accuracy level increased approximately by 07% in the qualitative tests. The average number of flexible features were added to the stylometric feature set was around 80 new features when the tests contains datasets that have size between 02 to 10 different persons. When the dataset contains up to 30 different persons, the average number of created flexible features was up to 420 new features. As a fact, the execution time of the prototype after applying the project Alpha was prolonged by 19% due to the increment in the processing time of the classifier when it trained the predicting model.

After the project Alpha, it can be observed that when the number of the stylometric features is increased by 50%, the accuracy level is improved by around 7% and resulted in prolonging the execution time by almost 20%. It is easy to see that the more features are used in the feature extractor will result in the higher accuracy level, but the increment in the accuracy level is much lower compares to the increment in the number of features. The above statistic cannot show the overall view about the relationship between the number of features and the accuracy level, since that relationship also depends on other factors, such as the quality of the stylometric feature set or the number of the available samples for the training process.

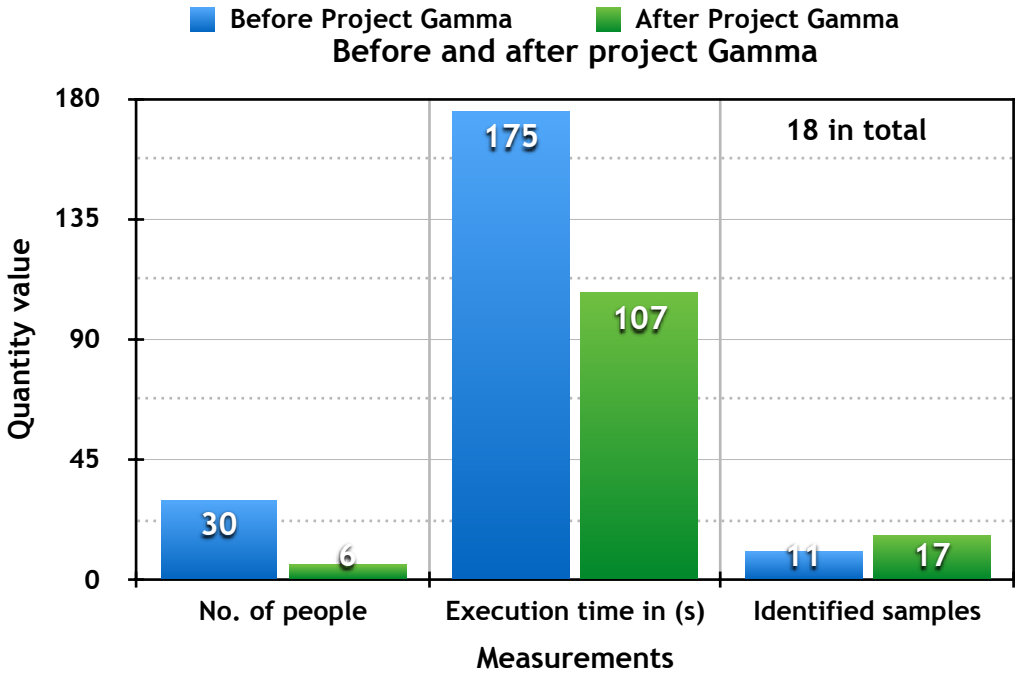
Due to the fact that project Alpha did help improving the qualitative result of the prototype although it is just a small improvement, we will still consider it as an useful enhancement for the initial prototype, thus in order to help other the improvement projects can maximize their abilities, the project Alpha will become a standard part of the initial prototype. It means that when other improvement projects are applied to the prototype, they will be applied on top the project Alpha. Since the project Alpha prolongs the execution time of the prototype. Thus, other improvement projects will also consider this problem when they suggest their own enhancements for the prototype.

5.6.3 Project Beta

Project Beta, with the support from the unsupervised learning technique, was able to improve the performance of the initial prototype. Although the qualitative attribute was only slightly improved, the quantitative attribute gained a notable result. The execution time reduced by almost 1/3 after the project Beta was applied to the prototype. During the experiments, for processing a dataset that contains ten different persons, the initial prototype took approximately 29 seconds to finish the authorship identification task while after the project Beta is applied, the prototype only took about 21 seconds to complete the same task. This is the result of shorter time of model training process by the classifier. Before the project Beta, the classifier must deal with a dataset of ten different persons in the training task. After the project Beta is applied, the number of different persons in the dataset was reduced by almost 60%.

From the quantitative experiments on the initial prototype with the support vector machine classifier from the previous chapter, the value of the execution time has an inclination to double when the number of different persons in the dataset is doubled. In this project, when the number of different persons in the dataset is reduced by half, the execution time is not also reduced by half, but by 1/3 only. The reason is that this extra amount of time is used in the clustering process, when the clusterer component tries to form groups of persons that share similarities in their writing styles.

5.6.4 Project Gamma



The project Gamma, with the idea of dividing the original dataset into smaller ones and individually working on them, gained notable results in improving the qualitative and quantitate values of the prototype. In the experiments with the project Gamma, the number of different persons in the dataset is 30. After the project Gamma is applied to the prototype, the accuracy level increased up to 35% of correctness, while the execution time is reduced by 40% for the qualitative and quantitative experiments on the same datasets. This is done by overcoming the weakness of the initial solution, which is the non-linear relationship between the number of different persons in the dataset and the prototype's accuracy level and its execution time in the authorship identification task. Similar to the project Beta, the number of different persons in the dataset that was given to the classifier was reduced by 40%, compares to the original size of the dataset when it is given to the prototype before the project Gamma is applied.

Similar to the project Beta, the variation of the execution time before and after the project Gamma is applied is not inclinable to be relational to the relationship between the execution time and the number of different persons in the dataset in the initial prototype. Bases on the statistic from the quantitative results of the initial prototype, the execution time of the prototype on a dataset that has 05 times larger in size than the another dataset will tend to result in 08 times longer in the execution time. Yet in this project, the execution time of the prototype after the project Gamma is applied is only smaller than the execution time of the initial prototype by 1.5 times. This extra amount of time is also the amount of time that the project Gamma needs to reduce the size of the original dataset. As a brief summary, the total time that the project Gamma needs to process, reduce the size of the original dataset and perform the authorship identification task on the new dataset is still lower than the execution time of processing the original dataset by the initial prototype by 1.5 times, while the accuracy level of the prototype after applying project Gamma is better than by 40% compares to before the project Gamma is applied.

5.6.5 Project Delta

The project Delta, with the support from the unsupervised learning technique in order to help to reduce the number of different persons in the dataset before it is given to the classifier, also gained notable results in improving the qualitative and quantitative attributes of the prototype. In the experiments with the project Gamma, the number of different persons in the dataset is ten. After the project Delta is applied, the accuracy level was increased by almost 10% during the qualitative experiments on the same dataset, while the execution time was almost reduced by a half compares to the execution time of the initial prototype. These achieved improvements were the result of the new smaller dataset which is used to give to the classifier. The new dataset was only haft of the size of the original dataset.

Unlike the project Beta and Gamma, the variation in the execution time before and after applying project Delta seems to be relational to the relationship between the execution time and the number of different persons in the dataset in the initial prototype. In the initial prototype, the dataset with ten different persons will usually result in the double of execution time compares to the dataset that only has five different persons. The project Delta was able to maintain this constraint between the number of different persons in the dataset and the execution time. The execution time of the prototype after applying the project Delta is only a half of the initial prototype's execution time, which corresponds to the decrement in the size of the old and new datasets. It seems that the project Delta did a better job than the project Beta and Gamma in the task of reducing the size of the original dataset.

VI. Conclusion and future work

In a general view, this thesis can be divided into two different parts. The first part was about evaluating the existing solutions in the authorship identification research area for their qualitative (the accuracy level) and quantitative (the performance time) attributes. While in the second part, the main objective was to determine the possible enhancements that could be applied to the initial solution in order to improve its performance.

The prototype was implemented in the first part of this thesis work as a solution for solving the authorship identification problem, bases on existing works from the other researchers. The implementation contains two main parts, the feature extractor with the stylometric feature set and the classifier. For the stylometric feature set, we did a selective selection on the proposed feature sets from different researchers and chose the ones that are suitable to our research purpose, then we used the combination of satisfied feature sets in the implementation of the feature extractor. In this thesis, we focused on developing a general solution which can work independently from the contents of the datasets, thus the stylometric features that belong to the content specific category were eliminated, the other feature categories that were used of are syntactic, structural and lexical. For the classifier component, we chose to implement the classifier with the neural network and the support vector machine techniques, that are the most frequently used classifying techniques by the other researchers. The comparison between these two classifying techniques for their qualitative and quantitative attributes were performed. In the collected results, the support vector machine classifier outperformed the neural network classifier in the qualitative experiment, although its performance in the quantitative experiment was slightly slower than the neural network classifier.

From the qualitative and quantitative experiments on the built prototype, we observed that when the number of different persons in the dataset is more than 30, there is a significant drop in the accuracy level. Furthermore, it seems that the execution time of the prototype is also doubled when the number of different persons in the dataset is doubled. Based on that observation, four improvement projects (Alpha, Beta, Gamma and Delta) were given with the expectation in improving the qualitative and quantitative results of the initial prototype. Based on that objective, we looked into how the existing solution works again. And from our perspective, we suggested that the qualitative and quantitative attributes can be improved by two factors, which are the quality of the stylometric feature set and the size of the dataset that is given to the prototype. Other factors such as improving current classifying algorithms or using other techniques that are different from the stylometric feature set to recognize the writing patterns is out this research's scope.

For the first factor, which is the quality of the stylometric feature set, we tried creating more features, besides the original ones, to improve the quality of the writing style seeking task. The project Alpha represents for that idea. In the project Alpha, we were able to create more features by using similar words in all of the training samples in the dataset. We called them flexible features because their contents would change when different datasets were used. When the project Alpha is applied to the initial prototype, we gained a good improvement in the qualitative attribute. The accuracy level increased up to 07%. The downside of this project was the increment in the execution time of the prototype.

For the second factor, which is reducing the size of the dataset before it is given to the classifier, our objective was to produce a better dataset that only contains highly relevant data, which is the most corresponding to the value of the testing samples. The project Beta, Gamma and Delta represent for that idea. The selectively reducing the dataset task is performed with two techniques, by making use of the unsupervised learning (clustering) and the supervised learning algorithms (classifying). For the first technique which is dividing the dataset into smaller parts by using a classifier, we changed the procedure of the original authorship identification task in order to fit our purpose. This idea is presented in the project Gamma. In the project Beta and Delta, we also did the same things we did in project Gamma, but instead of using classifier to divide the original dataset to reduce its size, we made use of the unsupervised learning technique in order to remove the irrelevant data in the dataset. The difference between project Gamma and the two other projects Beta and Delta is that, in project Gamma, at the end, all of the data in the dataset will be still processed by the classifier, while in the projects Beta and Delta, only data that is marked as highly relevant to the testing samples will be processed by the classifier. In the qualitative experiments, the project Gamma gained better accuracy level than the projects Beta and Delta, while in the quantitative experiments, projects Gamma and Delta outperformed the project Beta. The notable point is that although, in all of the Beta, Gamma and Delta projects, the size of the dataset was reduced significantly and only contained highly relevant data, project Beta was able to gain a better result in the qualitative experiments compares to the two others. The reason for this could belong to the structure of data in the dataset, which were decided by the classifying and the clustering techniques. More research works will be needed in order to understand this fact better.

In the Beta and Delta improvement projects, by making use of the unsupervised learning technique in different ways, the improvement projects were able to help to improve the qualitative and the quantitative attributes of the initial solution. Yet there is an important point when using the clustering technique, that is how to control the probability that the original writers of the testing samples are also appeared in the newly reduced dataset. This mainly depends on the ability of the clusterer, if the clusterer fails to recognize the correct group that the data should belong to, and the result that produced by the classifier later will only contain false negative values. In brief, the biggest challenge in the project Beta and Delta is that the clusterer must be good enough to avoid the false negative cases by only producing clusters that contain highly relevant data. More research works on the clustering technique will be needed in order to control the quality of the produced dataset better.

Besides the possible future works on analyzing the correlation between the structure of the dataset and the trained model that were produced by the classifier, or the controlling of the result's quality in clustering technique, which can help to extend the performance of the project Beta, Gamma and Delta. The gender detection technique can be considered as a new opportunity for improving the qualitative and the quantitative attributes of the authorship identification task.

Detecting writer's gender from their writings is not an illusory task. From the research works of Koppel, Argaamon and Shimoni [6.1] and Argamon, Saric and Stein [6.2], they provided some reliable evidences to prove that the writing style between a male and a female person is distinguishable by looking at the usage of the pronouns and the certain types of the noun modifier words. Gender detection technique can help to improve the qualitative and the quantitive attributes of the authorship analysis task by providing one more evaluating factors for comparing the correlation between the predicted writer of a testing sample with its original writer. The idea is simple, in order to improve the accuracy level of the authorship identification task, if the predicted gender of the predicted writer is the same as the predicted gender of the original writer of a given sample, there will be more confident in order to accept the predicted result, or vice versa. For improving the quantitative result of the authorship identification task, the gender detection technique can help to reduce the size of the original dataset by removing persons that do not have the same gender as the gender of the testing sample's original writer. This helps to reduce the works need to be done by the classifier. In brief, the gender detection technique is a promising opportunity in order to help to improve both of the qualitative and quantitative results of the authorship identification task. One of the most entangles here is about the ability of the gender detection technique since wrong results in predicted gender may result in a total failure in the classifying task later.

Another improvement for the authorship identification solution will be upgraded its current stylometric feature set so it can work with samples that have less than 250 words. Although as stated before, it is very difficult for attributing a writing that has less than 250 words to its original writer, it does not mean that this task is impossible. Zheng [2.3] claimed that although short online messages have their limitation in the word length, they still has other characteristics that may be able to help in revealing the writing style of the writer. For examples, the structure of short online writings is often different from the regular text documents, probably because of the different purposes in these two kinds of writing. By looking at some special features such as the structural layout traits or the unusually content markers of the short online texts, it can help to reduce the search range in the authorship identification task.

VII. References

[1.1] Holmes D., The evolution of stylometry in humanities scholarship, literary and linguistic computing, 1998, pp. 111-117.
[1.2] Smita Nirkhi and R.V.Dharaskar, Comparative study of authorship identification techniques for cyber forensics analysis, International journal of of advanced computer science and application, 2013, vol. 4, no. 5.
[1.3] O. De Vel et al., Mining email content for author identification forensics, 2001, vol. 30, no.4, pp. 55-64.
[1.4] J.F. Burrows, Word patterns and story shapes, the statistical analysis of narrative style, literary and linguistic computing, 1987, vol. 2, pp. 61-67.
[1.5] Ahmed Abbasi and Hsinchun Chen, Applying authorship analysis to extremist group web forum messages, 2005, IEEE magazine.
[1.6] R.Zheng et al., A framework of authorship identification for online message, writing style features and classification techniques, 2005.
[1.7] Rong Z, Jiexun L, Hsinchun C and Zan H. Writing-style features and classification techniques, 2005, Wiley Periodicals,Inc.
[1.8] Mealand D.L. The characteristic curves of composition, 1887, Science.
[1.9] Forsyth and Holmes, Feature finding for text classification, literary and linguistic computing, 1996.
[1.10] FBI, Cyber crime, http://www.fbi.gov/about-us/investigate/cyber

[2.1] M. Corney, A. Anderson, G. Mohay, Identifying the authors if suspect email, 2001, ACM magazine.
[2.2] O. de Vel, A. Anderson, M. Corney, G. Mohay, Mining email content for author identification forensics, 2001, Sigmod.
[2.3] Rong Zheng, Jiexun Li, HsinChun Chen, Zan Huang, A framework for authorship identification of online messages, writing style features and classification technique, 2005, Wiley InterScience.
[2.4] E. Stamatatos, N. Fakotakis, and G. Kokkinakis. Computer-based authorship attribution without lexical measures,2001, Computers and the Humanities, pp. 193–214.
[2.5] Yuta Tsuboi and Yuji Matsumoto, Authorship identification for heterogeneous documents, 2002, Nara Institute of Science and Technology, Japan.
[2.6] Na Cheng, R. Chandramouli, K.P. Subbalakshmi, Author identification from text, 2011, ScienceDirect.
[2.7] Ahmed Abbasi, Hsinchun Chen, Applying authorship analysis to extremist group web forum message, 2005, IEEE Computer Society.
[2.8] Neil Graham, Graeme Hirst, and Bhaskara Marthi, Segmenting documents by stylistic character, 2005, Nat. Lang. Eng.
[2.9] B. Kjell, Authorship determination using letter pair frequency features with neural network classifiers, 1994, In in Literacy and Linguistic Computing.
[2.10] J.F. Hoorn, S.L. Frank, W. Kowalczyk, and F. van der Ham. Neural network identification of poets using letter sequences, 1999, Lit Linguist Computing.
[2.11] J. Ross Quinlan, Induction of decision trees, 1986, Machine Learning.
[2.12] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten, The WEKA data mining software: An update, 2009. SIGKDD Explorations.

[2.13] J. B. MacQueen, Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, 1967, Berkeley, University of California Press.
[2.14] Shyam M. Guthikonda, Kohonen Self-Organizing Maps, 2005, Wittenberg University.
[2.15] David M. Blei, COS424 Hierarchical clustering, 2007, Princeton University.
[2.16] Osama Abu Abbas, Comparison between data clustering algorithms, 2007, Computer science department, Yarmouk university.
[2.17] Carl Staelin, Parameter selection for support vector machines, 2003, HP laboratories Israel.

[3.1] Na Cheng, R. Chandramouli, K.P. Subbalakshmi, Author identification from text, 2011, ScienceDirect.
[3.2] C. W. Hsu and C. J. Lin, A comparison of methods for multi-class support vector machines, 2002, Information engineering, National Taiwan University, IEEE Transactions on Neural Networks.
[3.3] T. Xie, H. Yu and B. Wilamowski, Comparison between traditional neural networks and radial basis function networks, Auburn University.
[3.4] Jeff Heaton, Programming neural networks with encog in Java, 2011, Heaton Research, Inc.
[3.5] Mehryar Mohri, Foundations of machine learning, 2012, MIT Press.
[3.6] Waren S. Sarle, Cary, Frequently asked question about neural network, 2002, www.fags.org.

[4.1] http://www.heatonresearch.com/encog
[4.2] http://www.csie.ntu.edu.tw/~cjlin/libsvm/
[4.3] http://java-ml.sourceforge.net/

[5.1] Michael Brennan, Sadia Afroz, and Rachel Greenstadt, Adversarial stylometry: Circumventing authorship recognition to preserve privacy and anonymity, 2012, ACM Trans. Inf. Syst. Secur. 15, 3, Article 12
[5.2] William W. Cohen, http://www.cs.cmu.edu/~./enron/, Carnegie Mellon University, accessed on March, 2014.
[5.3] Satoshi Nakamoto, Bit-coin: A peer to peer electronic cash system, https://bitcoin.org/bitcoin.pdf, accessed on March, 2014.
[5.4] Nermin Hajdarbegovic, Linguistic researchers name Nick Szabo as author of bit-coin whitepaper, http://www.coindesk.com/linguistic-researchers-name-nick-szabo-author-bitcoin-whitepaper/, accessed on April 2014.

[6.1] Koppel, M. Argamon and S. Shinomi, Automatically categorizing written texts by author gender, 2002, A.R Literary and linguistic computing.
[6.2] Argamon, S. Saric and M. Stein, Style mining of electronic messages for multiple authorship discrimination, 2003, Conference on knowledge discovery and data mining, ACM Press.

Appendix

In most of the case, the raw data provided below can be used in order to simulate the results again. The only exception is for the classifying tasks performed by the neural network classifier, which have different weight matrix because of randomly given weight at the beginning of training time.

A.1 Experiment with the support vector machine classifier

Test ID	Set	Number of ideal	Training samples	Testing samples	Accuracy	Performance	Type	Kernel	Cost	Gamma	Nu	Type of Classifier	IdealList
1	BG	2	23	6	100%	10133	SVM	Sigmoid	2	0,0078125	0,5	nu-SVC	e,t
2	BG	2	18	5	100%	7927	SVM	Sigmoid	128	1,12E-04	0,5	nu-SVC	a,e
3	BG	2	27	6	67%	8763	SVM	Sigmoid	2	0,0078125	0,5	nu-SVC	cc,ww
4	BG	3	38	9	89%	12295	SVM	Sigmoid	128	1,22E-04	0,5	nu-SVC	f,jj,q
5	BG	3	37	9	89%	11848	SVM	Sigmoid	128	1,22E-04	0,5	nu-SVC	c,j,rr
6	BG	3	38	9	77%	15152	SVM	Sigmoid	2	0,0078125	0,5	nu-SVC	cc,t,tt
7	BG	5	62	15	94%	14335	SVM	Sigmoid	2	0,0078125	0,5	nu-SVC	m,nn,qq,s,x
8	BG	5	72	15	80%	16557	SVM	Sigmoid	32	4,88E-04	0,5	nu-SVC	g,j,p,pp,rr
9	BG	5	59	15	80%	27687	SVM	Sigmoid	2	0,0078125	0,5	nu-SVC	bb,j,s,t,x
10	BG	10	126	30	83%	31079	SVM	Sigmoid	32	0,001953125	0,5	nu-SVC	aa,b,e,f,j,ii,k,kk,tt,u
11	BG	10	132	30	87%	26985	SVM	Sigmoid	512	1,22E-04	0,5	nu-SVC	aa,e,h,hh,j,o,pp,r,rr,ss
12	BG	10	120	30	77%	29587	SVM	Sigmoid	8	0,0078125	0,5	nu-SVC	c,cc,e,j,ll,nn,tt,vv,w,y
13	BG	30	370	89	60%	174060	SVM	Sigmoid	512	1,22E-04	0,5	nu-SVC	a,aa,b,bb,c,dd,f,ff,g,ii,jj,k,l,ll,m,mm,n,o,pp,q,qq,r,s,sst,tt,u,uu,y,z
14	BG	30	377	87	57%	146528	SVM	Sigmoid	8	0,0078125	0,4	nu-SVC	aa,b,bb,,c,cc,e,g,gg,h,hh,i,ii,jj,k,kk,ll,m,o,p,q,qq,r,s,ss,tt,vv,w,ww,z
15	BG	30	368	87	65%	141536	SVM	Sigmoid	8	0,0078125	0,3	nu-SVC	aa,b,bb,c,d,dd,e,ff,gg,hh,i,j,jj,kk,m,mm,p,pp,q,qq,r,rr,t,uu,w,ww,y,z
16	Enron	2	20	6	100%	8157	SVM	Sigmoid	2	0,0078125	0,5	nu-SVC	guzman-m,lokay-m
17	Enron	2	20	6	100%	7609	SVM	Sigmoid	0,03125	0,0078125	0,5	nu-SVC	davis-d,grigsby-m
18	Enron	2	20	6	84%	7926	SVM	Sigmoid	0,03125	0,0078125	0,5	nu-SVC	delainey-d,mims-thurston-p
19	Enron	3	30	9	89%	7947	SVM	Sigmoid	2	0,0078125	0,5	nu-SVC	daleiney-d,scott-s,taylor-m
20	Enron	3	30	9	89%	9711	SVM	Sigmoid	0,03125	0,0078125	0,5	nu-SVC	base-e,kuykendall-t,sanders-r
21	Enron	3	30	9	67%	8327	SVM	Sigmoid	2	0,0078125	0,5	nu-SVC	farmer-d,fossum-d,weldon-c
22	Enron	5	50	15	74%	16708	SVM	Sigmoid	2	0,0078125	0,5	nu-SVC	allen-p,dasovich-j,davis-d,germany-c,guzman-m
23	Enron	5	50	15	94%	12540	SVM	Sigmoid	128	1,22E-04	0,5	nu-SVC	cash-m,fossum-d,lokay-m,love-p,rusctitti-k
24	Enron	5	50	15	94%	16993	SVM	Sigmoid	8	0,0078125	0,5	nu-SVC	dorland-c,giron-d,kaminski-v,kean-s,symes-k,
25	Enron	10	100	30	77%	43953	SVM	Sigmoid	32	4,88E-04	0,5	nu-SVC	allen-p,campbell-l,fossum-d,giron-d,haedicke-m,horton-s,love-p,nemec-g,symes-k,taylor-m,
26	Enron	10	100	30	80%	25331	SVM	Sigmoid	512	1,22E-04	0,5	nu-SVC	beck-s,campbell-l,cash-m,dasovich-j,fossum-d,neal-s,nemec-g,rogers-b,shankman-j,weldon-c
27	Enron	10	100	30	73%	25169	SVM	Sigmoid	512	1,22E-04	0,5	nu-SVC	bass-e,germany-c,giron-d,haedicke-m,hain-m,mcconnell-m,rogers-b,shankman-j,taylor-m,weldon-c
28	Enron	30	300	90	54%	108249	SVM	Sigmoid	512	1,22E-04	0,5	nu-SVC	beck-s, campbell-l, cash-m, dasovich-j, davis-d, delainey-d, dickson-s, dorland-c, farmer-d, gay-r, grigsby-m, haedicke-m, hain-m, horton-s, jones-t, kaminski-v, kean-s, lavorato-j, lokay-m, mann-k, mclaughlin-e, neal-s, perlingiere-d, rogers-b, rusctitti-k, sager-e, scott-s, shankman-j, smith-m, taylor-m
29	Enron	30	300	90	54%	111914	SVM	Sigmoid	8	0,0078125	0,5	nu-SVC	allen-p, bass-e, beck-s, campbell-l, dasovich-j, davis-d, dorland-c, farmer-d, germany-c, guzman-m, hain-m, hyvl-d, kaminski-v, kean-s, lavorato-j, lenhart-m, love-p, mann-k, mcconnell-m, mclaughlin-e, nemec-g, pereira-s, perlingiere-d, rogers-b, rusctitti-k, sanders-r, scott-s, shankman-j, smith-m, tholt-j
30	Enron	30	300	90	52%	107380	SVM	Sigmoid	8	0,0078125	0,5	nu-SVC	allen-p, beck-s, cash-m, davis-d, dickson-s, fossum-d, gay-r, germany-c, grigsby-m, guzman-m, hain-m, horton-s, kaminski-v, kuykendall-t, lavorato-j, lokay-m, love-p, mann-k, mcconnell-m, mclaughlin-e, mims-thurston-p, neal-s, pereira-s, perlingiere-d, rusctitti-k, sager-e, scott-s, shackleton-s, shankman-j, taylor-m

A.2 Experiment with the neural network classifier

Test ID	Set	Number of Ideal	Training samples	Testing samples	Accuracy	Performance	Activation	Hidden layer	Hidden neutron	Iteration	Error Train/Test	IdealList
1	BG	2	27	6	100% | 67% | 83%	5446 | 3497 | 3403	Sigmoid	1	502	19 | 14 | 17	1.7E-6/0.02 | 5.2E-4/0.2 | 0.012/0.09	hh, t
2	BG	2	26	6	67% | 83% | 83%	5861 | 4437 | 1904	Sigmoid	1	502	18 | 19 | 11	1.6E-4/0.1 | 1.19E-6/0.09 | 2.4E-13/0.04	i, uu
3	BG	2	26	6	100% | 100% | 100%	9068 | 4651 | 9739	Sigmoid	1	502	11 | 15 | 23	9.4E-6/5.0E-41 | 1.52E-5/8.2E-24 | 0.009/1.9E-14	bb, x
4	BG	3	36	9	89% | 56% | 77%	10996 | 14462 | 4892	Sigmoid	2	252	40 | 59 | 41	3.9E-4/0.07 | 9.5E-6/0.2 | 4.3E-5/0.14	b, j, nn
5	BG	3	39	9	77% | 77% | 56%	10323 | 4663 | 4317	Sigmoid	2	252	40 | 42 | 40	4.5E-4/0.07 | 2.4E-4/0.14 | 0.01/0.19	h, kk, x
6	BG	3	38	9	77% | 67% | 56%	11014 | 5124 | 5939	Sigmoid	2	252	44 | 46 | 48	0.009/0.08 | 0.01/0.1 | 0.01/0.17	aa, c, q
7	BG	5	65	15	47% | 54% | 60%	14422 | 7317 | 7084	Sigmoid	2	254	52 | 52 | 54	0.01/0.14 | 0.004/0.14 | 0.002/0.12	dd, hh, m, s, uu
8	BG	5	79	15	47% | 54% | 54%	15535 | 9632 | 9393	Sigmoid	2	254	59 | 62 | 60	0.007/0.19 | 0.007/0.15 | 0.01/0.15	j, k, pp, qq, u
9	BG	5	57	15	67% | 93% | 60%	14601 | 7150 | 5999	Sigmoid	2	254	52 | 58 | 41	0.01/0.07 | 0.003/0.01 | 0.002/0.14	g, ii, l, rr, s
10	BG	10	119	30	47% | 33% | 47%	16707 | 14409 | 11562	Sigmoid	3	172	48 | 60 | 48	0.006/0.08 | 0.006/0.08 | 0.009/0.06	d, ff, jj, k, l, ll, m, p, t, w
11	BG	10	116	30	30% | 43% | 43%	28905 | 17258 | 18342	Sigmoid	2	259	114 | 77 | 81	0.008/0.09 | 0.005/0.07 | 0.007/0.07	d, e, nn, s, ss, t, u, uu, w, ww
12	BG	10	126	30	60% | 47% | 50%	22262 | 14595 | 20923	Sigmoid	2	259	66 | 55 | 79	0.01/0.05 | 0.009/0.07 | 0.006/0.07	aa, c, cc, d, ii, kk, o, p, w, z
13	BG	30	359	90	3% | 12% | 9%	61397 | 67195 | 63427	Sigmoid	2	279	58 | 57 | 65	0.007/0.03 | 0.007/0.03 | 0.008/0.03	aa, b, d, dd, e, f, ff, g, gg, hh, i, ii, j, jj, k, l, m, mm, n, nn, o, p, r, rr, ss, tt, u, vv, x, y
14	BG	30	359	89	18% | 16% | 19%	71709 | 66172 | 72094	Sigmoid	1	502	20 | 12 | 12	1.2E-14/0.16 | 5.3E-4/0.08 | 3.5E-8/1.8E-40	a, aa, b, cc, d, e, f, ff, g, gg, h, ii, j, k, kk, l, m, nn, o, qq, r, s, ss, t, vv, w, ww, x, y, z
15	BG	30	363	89	20% | 7% | 21%	56333 | 48288 | 48765	Sigmoid	2	279	65 | 65 | 55	0.009/0.03 | 0.009/0.03 | 0.008/0.03	a, b, d, dd, e, f, ff, g, gg, h, j, kk, l, ll, mm, n, o, p, q, qq, r, rr, s, ss, u, uu, w, ww, x, z
16	Enron	2	20	6	100% | 67% | 67%	5427 | 3870 | 2216	Sigmoid	1	502	10 | 22 | 16	2.1E-6/2.1E-8 | 2.3E-8/0.15 | 0.01/0.34	hain-m, rusciti-k
17	Enron	2	20	6	67% | 83% | 100%	5812 | 3014 | 3037	Sigmoid	1	502	20 | 12 | 12	1.2E-14/0.16 | 5.3E-4/0.08 | 3.5E-8/1.8E-40	cash-m, grigsby-m
18	Enron	2	20	6	67% | 83% | 83%	5046 | 3117 | 3825	Sigmoid	1	502	11 | 10 | 17	1.1E-4/0.11 | 1.8E-4/0.11 | 7.4E-5/0.06	allen-p, rogers-b
19	Enron	3	30	9	56% | 67% | 56%	6509 | 6090 | 3931	Sigmoid	2	252	49 | 39 | 43	1.4E-9/0.2 | 0.004/0.2 | 0.019/0.18	horton-s, lenhart-m, pereira-s
20	Enron	3	30	9	89% | 56% | 77%	13963 | 11617 | 8522	Sigmoid	2	252	43 | 35 | 50	3.2E-12/0.015 | 0.01/0.1417 | 6E-10/0.14	campbell-l, lokay-m, rogers-b
21	Enron	3	30	9	67% | 77% | 45%	7713 | 4753 | 3570	Sigmoid	2	252	56 | 44 | 41	0.02/0.18 | 5.5E-7/0.11 | 1.9E-40.31	campbell-l, horton-s, pereira-s
22	Enron	5	50	15	60% | 60% | 47%	15848 | 8156 | 7619	Sigmoid	2	254	52 | 69 | 70	0.006/0.12 | 0.003/0.14 | 0.01/0.15	germany-c, giron-d, mann-k, mims-thurston-p, sager-e
23	Enron	5	50	15	60% | 60% | 47%	17235 | 7611 | 8376	Sigmoid	2	254	51 | 48 | 48	0.007/0.14 | 0.006/0.11 | 0.011/0.16	davis-d, delainey-d, dorland-c, rusciti-k, scott-s
24	Enron	5	50	15	60% | 60% | 60%	12454 | 5715 | 5551	Sigmoid	2	254	50 | 45 | 49	0.01/0.17 | 0.01/0.12 | 0.005/0.1	campbell-l, cash-m, giron-d, horton-s, shackleton-s
25	Enron	10	100	30	27% | 43% | 47%	20968 | 13051 | 12528	Sigmoid	2	259	89 | 74 | 67	0.005/0.12 | 0.005/0.08 | 0.007/0.08	dorland-c, farmer-d, fossum-d, mims-thurston-p, neal-s, pereira-s, rogers-b, shankman-j, tholt-j, weldon-c
26	Enron	10	100	30	43% | 47% | 50%	42579 | 13582 | 10508	Sigmoid	2	259	260 | 73 | 50	0.015/0.07 | 0.009/0.06 | 0.005/0.06	beck-s, davis-d, gay-r, hyvl-d, kaminski-v, pereira-s, perlingiere-d, rusciti-k, shackleton-s, shankman-j
27	Enron	10	100	30	30% | 37% | 30%	21064 | 14128 | 13577	Sigmoid	2	259	86 | 84 | 79	0.008/0.1 | 0.007/0.08 | 0.007/0.09	delainey-d, farmer-d, fossum-d, germany-c, hyvl-d, lavorato-j, mann-k, shackleton-s, taylor-m, tholt-j
28	Enron	30	300	90	12% | 9% | 9%	164306 | 59886 | 49542	Sigmoid	2	279	381 | 83 | 73	0.01/0.05 | 0.008/0.03 | 0.01/0.03	arnold-j, bass-e, beck-s, dasovich-j, davis-d, delainey-d, dickson-s, farmer-d, fossum-d, gay-r, germany-c, giron-d, guzman-m, hain-m, jones-t, kaminski-v, kean-s, lavorato-j, lenhart-m, love-p, mann-k, mcconnell-m, mims-thurston-p, nemec-g, pereira-s, sager-e, scott-s, smith-m, symes-k, weldon-c
29	Enron	30	300	90	12% | 22% | 12%	43214 | 43532 | 40968	Sigmoid	2	279	76 | 85 | 78	0.009/0.03 | 0.01/0.03 | 0.008/0.03	arnold-j, bass-e, campbell-l, cash-m, dasovich-j, davis-d, delainey-d, dorland-c, farmer-d, fossum-d, gay-r, giron-d, guzman-m, haedicke-m, horton-s, hyvl-d, jones-t, kaminski-v, kuykendall-t, lavorato-j, lenhart-m, lokay-m, mann-k, mcconnel-m, mclaughlin-e, neal-s, rogers-b, sanders-r, smith-m, symes-k
30	Enron	30	300	90	16% | 13% | 7%	45505 | 43277 | 281600	Sigmoid	2	279	81 | 86 | 667	0.008/0.03 | 0.01/0.03 | 0.01/0.05	arnold-j, bass-e, davis-d, dickson-s, farmer-d, fossum-d, germany-c, giron-d, grigsby-m, hyvl-d, jones-t, kean-s, kuykendall-t, lenhart-m, lokay-m, love-p, mann-k, mims-thurston-p, nemec-g, pereira-s, sager-e, scott-s, shackleton-s, shankman-j, smith-m, symes-k, taylor-m, tholt-j, weldon-c

A.3 Project Alpha

Test ID	Set	Number of Ideal	Training samples	Testing samples	Extra feature	Accuracy	Performance	Type	Kernel	Cost	Gamma	Nu	Type of Classifier	Ideal.list
1	BG	2	23	6	28	100%	7900	SVM	Sigmoid	128	1,12E-04	0,5	nu-SVC	e,t
2	BG	2	18	5	22	80%	7521	SVM	Sigmoid	128	1,12E-04	0,5	nu-SVC	a,e
3	BG	2	27	6	33	84%	9052	SVM	Sigmoid	128	1,12E-04	0,5	nu-SVC	cc,ww
4	BG	3	38	9	47	89%	11808	SVM	Sigmoid	2	7,81E-03	0,5	nu-SVC	f,jj,q
5	BG	3	37	9	46	89%	10254	SVM	Sigmoid	2	7,81E-03	0,5	nu-SVC	c,i,rr
6	BG	3	38	9	47	100%	11186	SVM	Sigmoid	128	1,12E-04	0,5	nu-SVC	cc,t,tt
7	BG	5	62	15	77	100%	14911	SVM	Sigmoid	2	7,81E-03	0,5	nu-SVC	m,nn,qq,s,x
8	BG	5	72	15	87	74%	16222	SVM	Sigmoid	512	1,12E-04	0,5	nu-SVC	g,j,p,pp,rr
9	BG	5	59	15	74	87%	14756	SVM	Sigmoid	128	1,12E-04	0,5	nu-SVC	bb,j,s,t,x
10	BG	10	126	30	159	77%	34673	SVM	Sigmoid	2	7,81E-03	0,5	nu-SVC	aa,b,e,f,i,ii,k,kk,tt,u
11	BG	10	132	30	164	90%	35106	SVM	Sigmoid	512	1,12E-04	0,5	nu-SVC	aa,e,h,hh,j,o,pp,r,rss
12	BG	10	120	30	150	64%	31391	SVM	Sigmoid	8	7,81E-03	0,5	nu-SVC	c,cc,e,j,ll,nn,tt,vv,w,y
13	BG	30	370	89	431	62%	167453	SVM	Sigmoid	32	1,95E-03	0,4	nu-SVC	a,aa,b,bb,c,dd,f,ff,g,j,jj,k,l,ll,m,mm,n,o,pp,q,qq,r,s,sst,tt,u,uu,y,z
14	BG	30	377	87	466	64%	193044	SVM	Sigmoid	32	1,95E-03	0,4	nu-SVC	aa,b,bb,,c,cc,e,g,gg,h,hh,j,ii,jj,k,kk,ll,m,o,p,q,qq,r,s,ss,tt,vv,w,ww,z
15	BG	30	368	87	442	74%	179365	SVM	Sigmoid	512	1,22E-04	0,3	nu-SVC	aa,b,bb,c,d,dd,e,ff,gg,hh,j,jj,jjj,kk,m,mm,p,pp,q,qq,r,rr,t,uu,w,ww,y,z
16	Enron	2	20	6	26	100%	8205	SVM	Sigmoid	2	0,0078125	0,5	nu-SVC	guzman-m,lokay-m
17	Enron	2	20	6	26	100%	8125	SVM	Sigmoid	0,003125	0,0078125	0,5	nu-SVC	davis-d,grigsby-m
18	Enron	2	20	6	26	84%	8015	SVM	Sigmoid	0,003125	0,0078125	0,5	nu-SVC	delainey-d,mims-thurston-p
19	Enron	3	30	9	26	100%	8233	SVM	Sigmoid	0,003125	0,0078125	0,5	nu-SVC	dalelney-d,scott-s,taylor-m
20	Enron	3	30	9	26	100%	8180	SVM	Sigmoid	0,003125	0,0078125	0,5	nu-SVC	base-e,kuykendall-t,sanders-r
21	Enron	3	30	9	39	89%	9292	SVM	Sigmoid	8	0,001953125	0,5	nu-SVC	farmer-d,fossum-d,weldon-c
22	Enron	5	50	15	65	87%	12145	SVM	Sigmoid	2	0,0078125	0,5	nu-SVC	allen-p,dasovich-j,davis-d,germany-c,guzman-m
23	Enron	5	50	15	65	80%	12600	SVM	Sigmoid	2	0,0078125	0,5	nu-SVC	cash-m,fossum-d,lokay-m,love-p,ruscitti-k
24	Enron	5	50	15	65	100%	11636	SVM	Sigmoid	128	1,22E-04	0,5	nu-SVC	dorland-c,giron-d,kaminski-v,kean-s,symes-k
25	Enron	10	100	30	130	80%	26337	SVM	Sigmoid	8	7,81E-03	0,5	nu-SVC	allen-p,campbell-l,fossum-d,giron-d,haedicke-m,horton-s,love-p,nemec-g,symes-k,taylor-m
26	Enron	10	100	30	130	77%	26449	SVM	Sigmoid	128	1,22E-04	0,5	nu-SVC	beck-s,campbell-l,cash-m,dasovich-j,fossum-d,neal-s,nemec-g,rogers-b,shankman-j,weldon-c
27	Enron	10	100	30	120	87%	26793	SVM	Sigmoid	128	1,22E-04	0,5	nu-SVC	bass-e,germany-c,giron-d,haedicke-m,hain-m,mcconnell-m,rogers-b,shankman-j,taylor-m,weldon-c
28	Enron	30	300	90	390	58%	130032	SVM	Sigmoid	512	1,22E-04	0,5	nu-SVC	beck-s,campbell-l,cash-m,dasovich-j,delainey-d,dickson-s,dorland-c,farmer-d,gay-r,grigsby-m,haedicke-m,hain-m,horton-s,jones-t,kaminski-v,kean-s,lavorato-j,lokay-m,mann-k,mclaughlin-e,neal-s,pereira-s,perlingiere-d,rogers-b,ruscitti-k,sager-e,scott-s,shankman-j,smith-m,taylor-m
29	Enron	30	300	90	390	60%	136523	SVM	Sigmoid	8	0,0078125	0,5	nu-SVC	allen-p,bass-e,beck-s,campbell-l,dasovich-j,davis-d,dorland-c,farmer-d,germany-c,guzman-m,hain-m,hyvl-d,kaminski-v,kean-s,lavorato-j,lenhart-m,love-p,mann-k,mcconnell-m,mclaughlin-e,nemec-g,pereira-s,perlingiere-d,rogers-b,ruscitti-k,sanders-r,scott-s,shankman-j,smith-m,tholt-j
30	Enron	30	300	90	390	57%	133119	SVM	Sigmoid	8	0,0078125	0,5	nu-SVC	allen-p,beck-s,cash-m,davis-d,dickson-s,fossum-d,gay-r,germany-c,grigsby-m,guzman-m,hain-m,horton-s,kaminski-v,kuykendall-t,lavorato-j,lokay-m,love-p,mann-k,mcconnell-m,mclaughlin-e,mims-thurston-p,neal-s,pereira-s,perlingiere-d,ruscitti-k,sager-e,scott-s,shackleton-s,shankman-j,taylor-m

A.4 Project Beta

Type	Set	Accracy	Performance	Ideal	Test samples	UL Set	Flexible feature	c/gamma	Train samples
Project Beta	BG	100%	29538	10	i	[dd, nn, gg, i] | [i]	46	2/0.0078125	[cc, d, dd, gg, i, ll, nn, o, s, uu, unknown]
Normal	BG	100%	28911	10	i	[]	146	128.0/1.220703125 E-4	[cc, d, dd, gg, i, ll, nn, o, s, uu]
Project Beta	BG	67%	21167	10	t	[t, ll, p, ii, aa, jj] | [t]	74	128/1.22E-4	[aa, ii, jj, ll, m, mm, nn, p, t, w, unknown]
Normal	BG	67%	29081	10	t	[]	144	8.0/0.0078125	[aa, ii, jj, ll, m, mm, nn, p, t, w]
Project Beta	BG	100%	11056	10	ff	[w, ff, p] | [ff]	42	128/1.22E-4	[e, ff, g, i, p, r, rr, uu, w, ww, unknown]
Normal	BG	100%	32129	10	ff	[]	155	8.0/0.0078125	[e, ff, g, i, p, r, rr, uu, w, ww]
Project Beta	Enron	67%	24677	10	davis-d	[allen-p, davis-d, gay-r, campbell-l] | [davis-d]	43	128/1.22E-4	[allen-p, campbell-l, davis-d, dickson-s, dorland-c, gay-r, hyvl-d, kean-s, ruscitti-k, sager-e, unknown]
Normal	Enron	34%	23880	10	davis-d	[]	130	2/0.0078125	[allen-p, campbell-l, davis-d, dickson-s, dorland-c, gay-r, hyvl-d, kean-s, ruscitti-k, sager-e]
Project Beta	Enron	100%	20995	10	smith-m	[giron-d, smith-m] | [smith-m]	23	2/0.0078125	[bass-e, delainey-d, giron-d, hyvl-d, lavorato-j, mann-k, rogers-b, smith-m, taylor-m, tholt-j, unknown]
Normal	Enron	67%	24318	10	smith-m	[]	130	8.0/0.0078125	[bass-e, delainey-d, giron-d, hyvl-d, lavorato-j, mann-k, rogers-b, smith-m, taylor-m, tholt-j]
Project Beta	Enron	67%	23281	10	cash-m	[giron-d, shackleton-s, cash-m, mims-thurston-p, sanders-r, campbell-l, mclaughlin-e, kaminski-v] | [cash-m]	83	8/0.0078125	[campbell-l, cash-m, gay-r, giron-d, kaminski-v, mclaughlin-e, mims-thurston-p, ruscitti-k, sanders-r, shackleton-s, unknown]
Normal	Enron	67%	27167	10	cash-m	[]	130	8/0.0078125	[campbell-l, cash-m, gay-r, giron-d, kaminski-v, mclaughlin-e, mims-thurston-p, ruscitti-k, sanders-r, shackleton-s]

A.5 Project Gamma

Type	Set	Accracy	Performance	Ideal	Test samples	Final group FG	FG's Feature	FG's c/gamma	Train samples
ProjectGamma	BG	100%	125938	30	y	[c, rr, t, y, jj, cc, h]] [y]	94	128.0/1.2207031250E-4	[[aa, c, cc, d, dd, f, ff, g, h, i]]h,cc,c=y], [ij, k, kk, l, mm, n, nn, p, pp, rr]][rr,l,j]=y], [s, t, tt, u, uu, vv, w, x, y, z]][y=y]]
Normal	BG	67%	213902	30	y	[]	375	8.0/0.0078125	dd,ff,jj,nn,pp,rr,tt,vv,aa,cc,kk,mm,uu,c,d,f,g,h,i,k,l,n,p,s,t,u,w,x,y,z
ProjectGamma	BG	100%	115131	30	i	[u, s, rr, o, k, i]] [i]	76	128.0/1.2207031250E-4	[[a, aa, b, cc, e, ff, g, gg, h, i]][i=i], [j, jj, k, kk, l, m, n, nn, o, p]][o,k=i], [q, qq, r, rr, s, ss, u, uu, vv, y]][u,rr,s=i]]
Normal	BG	67%	213395	30	i	[]	374	512.0/1.2207031250E-4	ff,jj,nn,rr,vv,aa,cc,gg,kk,qq,ss,uu,a,b,e,g,h,i,j,k,l,m,n,o,p,q,rr,s,u,y
ProjectGamma	BG	100%	127094	30	r	[dd, tt, r, ii]] [r]	49	128.0/1.2207031250E-4	[[a, b, c, dd, e, f, ff, g, ii, jj]][dd,i=r], [k, kk, l, n, nn, o, p, pp, q, r]][r=r], [rr, s, ss, tt, u, uu, vv, www, y, z]][tt=r]]
Normal	BG	100%	210266	30	r	[]	371	8.0/0.0078125	dd,ff,nn,pp,rr,tt,vv,ii,kk,ss,uu,www,a,b,c,e,f,g,j,k,l,n,o,p,q,rs,u,yz
ProjectGamma	Enron	100%	90871	30	germany-c	[delainey-d, dasovich-j, smith-m, germany-c, lavorato-j, scott-s, hyvl-d, sanders-r] [germany-c]	83	128.0/1.2207031250E-4	[[arnold-j, dasovich-j, davis-d, delainey-d, dickson-s, dorland-c, farmer-d, fossum-d, germany-c, giron-d][germany-c, dasovich-j, delainey-d, germany-c, guzman-m, horton-s, hyvl-d, jones-t, lavorato-j, lenhart-m, mcconnell-m, mclaughlin-e, neal-s, pereira-s][lavorato-j, hyvl-d, germany-c], [rogers-b, ruscitti-k, sager-e, sanders-r, scott-s, shackleton-s, shankman-j, smith-m, symes-k, tholt-j]][scott-s,smith-m,sanders-r=germany-c]]
Normal	Enron	34%	139704	30	germany-c	[]	303	512.0/1.2207031250E-4	guzman-m,symes-k,hyvl-d,mcconnell-m,smith-m,fossum-d,giron-d,rogers-b,sager-e,dickson-s,farmer-d,lenhart-m,arnold-j,shankman-j,ruscitti-k,sanders-r,jones-t,horton-s,shackleton-s,dasovich-j,lavorato-j,scott-s,delainey-d,dorland-c,pereira-s,nemec-g,germany-c,tholt-j,mclaughlin-e,davis-d
ProjectGamma	Enron	67%	90022	30	mann-k	[sager-e, mann-k, shankman-j, arnold-j, sanders-r, campbell-l] [mann-k]	63	128.0/1.2207031250E-4	[[arnold-j, campbell-l, davis-d, delainey-d, dickson-s, dorland-c, farmer-d, fossum-d, grigsby-m, guzman-m][arnold-j,campbell-l,mann-k], [haedicke-m, kaminski-v, kean-s, kuykendall-t, lavorato-j, lenhart-m, mann-k, mclaughlin-e, neal-s, nemec-g][mann-k=mann-k], [perlingiere-d, rogers-b, ruscitti-k, sager-e, sanders-r, scott-s, shackleton-s, shankman-j, smith-m, tholt-j]][shankman-j,sanders-r,sager-e=mann-k]]
Normal	Enron	100%	135466	30	mann-k	[]	303	32.0/0.001953125	kuykendall-t,guzman-m,grigsby-m,campbell-l,smith-m,fossum-d,rogers-b,sager-e,kaminski-v,dickson-s,farmer-d,lenhart-m,arnold-j,perlingiere-d,shankman-j,ruscitti-k,sanders-r,shackleton-s,lavorato-j,scott-s,delainey-d,dorland-c,haedicke-m,mann-k,nemec-g,tholt-j,mclaughlin-e,neal-s,kean-s,davis-d
ProjectGamma	Enron	100%	95059	30	lavorato-j	[dorland-c, allen-p, shankman-j, lavorato-j, scott-s, arnold-j, kuykendall-t] [lavorato-j]	73	8.0/0.0078125	[[allen-p, arnold-j, base-e, dasovich-j, davis-d, dickson-s, dorland-c, farmer-d, gay-r, giron-d][arnold-j,dorland-c,allen-p,lavorato-j], [haedicke-m, hain-m, horton-s, jones-t, kuykendall-t, lavorato-j, lenhart-m, lokay-m, mann-k, mcconnell-m][kuykendall-t,lavorato-j=lavorato-j], [mclaughlin-e, mims-thurston-p, neal-s, nemec-g, ruscitti-k, scott-s, shankman-j, symes-k, taylor-m, weldon-c]][scott-s,shankman-j=lavorato-j]]
Normal	Enron	0%	136131	30	lavorato-j	[]	303	2.0/0.03125	dorland-c,kuykendall-t,mann-k,kuykendall-t,lokay-m,allen-p,lenhart-m,giron-d,arnold-j,symes-k,gay-r,taylor-m,neal-s,weldon-c,horton-s,base-e,hain-m,mcconnell-m,arnold-j,jones-t,farmer-d,shankman-j,lavorato-j,dickson-s,lavorato-j,shankman-j,scott-s,dasovich-j,davis-d,dorland-c,scott-s,nemec-g,ruscitti-k,haedicke-m,mclaughlin-e,mims-thurston-p

A.6 Project Delta

Type	Set	Ideal	Test samples	Accracy	Performance	Clustures	Union value	Filter clustured	Chosen randomly from clustured	Result from 1st SL	Formalizing final clustures	Feature	c and gamma
Project Delta	BG	10	r	100%	20432	[[e, gg, n, o, aa, j, h, jj], [vv, n, aa, j], [vv, e, r, gg, n, o, aa, j, jj]]	[n, aa, j]	[[e, gg, o, h, jj], [vv, e, r, gg, o, jj], [vv, n, aa, jj]]	[h, r, n]	[r]	[vv, e, r, gg, o, jj]	73	2.0/0.0078125
Normal	BG	10	r	100%	30625	gg, aa, jj, vv, r, e, h, j, n, o	[]	[]	[]	[]	[]	123	8.0/0.0078125
Project Delta	BG	10	ss	100%	13268	[[w, dd, ww, b, nn, ss, bb, j, kj], [dd, b, nn, bb, kk, kj], [w, dd, ww, nn, bb, j, kk, kj]]	dd, nn, bb, k]	[[w, ww, b, ss, j], [b, kk], [w, ww, j, kk], [dd, nn, bb, kj]]	[ss, dd]	[ss]	[w, ww, b, ss, j]	64	8.0/0.0078125
Normal	BG	10	ss	67%	27763	[b, bb, dd, j, k, kk, nn, ss, w, ww]	[]	[]	[]	[]	[]	158	128.0/1.220703125E-4
Project Delta	BG	10	l	67%	12894	[[u, ww, ff, uu, rr, o, aa, l, z, xj], [u, ff, l], [u, ww, ff, uu, rr, o, aa, l, z, xj]]	[u, ff, l]	[[ww, uu, rr, o, aa, z, xj], [ww, uu, rr, o, aa, z, xj], [u, ff, l]]	[u]	[u]	[u, ff, l]	37	2.0/0.0078125
Normal	BG	10	l	67%	32235	[aa, ff, l, o, rr, u, uu, ww, x, z]	[]	[]	[]	[]	[]	145	2.0/0.0078125
Project Delta	Enron	10	kuykendall-t	100%	11649	[[giron-d, dasovich-j, dorland-c, ruscitti-k, jones-t, mcconnell-m, lokay-m, kuykendall-t, love-p], [giron-d, dasovich-j, dorland-c, mims-thurston-p, jones-t, kuykendall-t], [giron-d, dorland-c, ruscitti-k, mims-thurston-p, mcconnell-m, lokay-m, kuykendall-t], [love-p]]	[giron-d, dasovich-j, dorland-c, kuykendall-t]	[[dasovich-j, ruscitti-k, jones-t, mcconnell-m, lokay-m, love-p], [dasovich-j, mims-thurston-p, jones-t, ruscitti-k, mims-thurston-p, mcconnell-m, lokay-m, love-p], [giron-d, dorland-c, kuykendall-t]]	[giron-d]	[giron-d, dorland-c, kuykendall-t]	33	2.0/0.0078125
Normal	Enron	10	kuykendall-t	100%	27498	kuykendall-t, dorland-c, ruscitti-k, jones-t, dasovich-j, mcconnell-m, mims-thurston-p, love-p, lokay-m, giron-d	[]	[]	[]	[]	[]	13	0.03125/0.0078125
Project Delta	Enron	10	cash-m	67%	13256	[[cash-m, shackleton-s, ruscitti-k, hain-m, shankman-j, arnold-j, pereira-s], [cash-m, shackleton-s, nemec-g], [cash-m, ruscitti-k, shankman-j, arnold-j, pereira-s], [cash-m, ruscitti-k, mims-thurston-p, shankman-j, arnold-j, pereira-s, mclaughlin-e, nemec-g]]	[cash-m, ruscitti-k, shankman-j, arnold-j, pereira-s]	[[shackleton-s, hain-m, mclaughlin-e, nemec-g], [mims-thurston-p, ruscitti-k, nemec-g], [shackleton-s, cash-m, ruscitti-k, shankman-j, arnold-j, pereira-s]]	[hain-m, mims-thurston-p, cash-m]	[cash-m, hain-m]	[cash-m, shackleton-s, ruscitti-k, hain-m, shankman-j, arnold-j, pereira-s, mclaughlin-e, nemec-g]	93	8.0/0.0078125
Normal	Enron	10	cash-m	67%	20473	[arnold-j, cash-m, hain-m, mclaughlin-e, mims-thurston-p, nemec-g, pereira-s, ruscitti-k, shackleton-s, shankman-j]	[]	[]	[]	[]	[]	130	32.0/4.8828125E-4
Project Delta	Enron	10	allen-p	67%	10487	[[cash-m, kean-s, perlingiere-d, weldon-c, farmer-d, lokay-m, dickson-s, pereira-s, fossum-d], [allen-p, kean-s, perlingiere-d, weldon-c, farmer-d, lokay-m, dickson-s], [cash-m, allen-p, kean-s, perlingiere-d, weldon-c, farmer-d, lokay-m, dickson-s, pereira-s, fossum-d]]	[weldon-c, perlingiere-d, kean-s, farmer-d, lokay-m, dickson-s]	[[cash-m, pereira-s, fossum-d], [cash-m, allen-p, pereira-s, fossum-d], [allen-p, weldon-c, perlingiere-d, kean-s, farmer-d, lokay-m, dickson-s]]	[kean-s]	[kean-s]	[allen-p, kean-s, perlingiere-d, weldon-c, farmer-d, lokay-m, dickson-s]	73	2.0/0.0078125
Normal	Enron	10	allen-p	67%	20228	[allen-p, cash-m, dickson-s, farmer-d, fossum-d, kean-s, lokay-m, pereira-s, perlingiere-d, weldon-c]	[]	[]	[]	[]	[]	130	512.0/1.220703125E-4

The end.