



**ANALYSIS OF MACHINE LEARNING – BASED SPAM FILTERING
TECHNIQUES**



NAZLI NAZLI

FEBRUARY 2018

**ANALYSIS OF MACHINE LEARNING – BASED SPAM FILTERING
TECHNIQUES**

**A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES OF
ÇANKAYA UNIVERSITY**

**BY
NAZLI NAZLI**

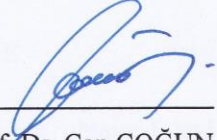
**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF
COMPUTER ENGINEERING**

FEBRUARY 2018

Title of the Thesis: **Analysis of machine learning-based spam filtering techniques**


Submitted by **Nazlı NAZLI**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University.


Prof. Dr. Can ÇOĞUN


Director


I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.


Prof. Dr. Erdoğan DOĞDU

Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.


Assist. Prof. Dr. Roya CHOUPANI
Co-Supervisor


Prof. Dr. Erdoğan DOĞDU
Supervisor

Examination Date: 09.02.2018
Examining Committee Members

Assoc. Prof. Tansel Özyer

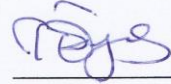
(TOBB ETU)

Prof. Dr. Erdoğan Dođdu

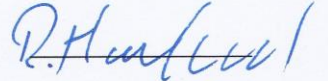
(Çankaya Univ.)

Assoc. Prof. Reza Zare Hassanpour

(Çankaya Univ.)








STATEMENT OF NON-PLAGIARISM PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Nazlı NAZLI

Signature : 

Date : 09.03.2018

ABSTRACT

Analysis of machine learning-based spam filtering techniques

NAZLI, Nazlı

M.Sc., Department of Computer Engineering

Supervisor: Prof. Dr. Erdoğan DOĞDU

Co-Supervisor: Assist. Prof. Dr. Roya CHOUPANI

February 2018, 79 pages

In this thesis, automatic spam e-mail detection problem is examined. Some existing machine learning algorithms are tested on an open dataset and the results are analyzed. The methods we developed have been implemented using machine learning and text classification techniques. We have used different data sets to develop and test the methods. The proposed methods for solving the problem are based on using weighted TF-IDF, SciKit Learn and Word2Vec vectorization. We developed and used vector representation methods for email text and then used supervised machine learning algorithms to classify emails as spam or ham. We used WEKA software tool to apply machine learning classification methods on vector representations of email. For classifications, we used the algorithms Support Vector Mechanism SVM (POLY), SVM (RBF), Naive Bayes, Bayesian Networks, J48 and Random Forest algorithms. We compared and analyzed the results we obtained from the classification methods. Our results show that the Word2Vec vector and the SVM (poly) algorithm perform better with 98.33% spam detection accuracy for 300 email data set.

Keywords: Spam emails, Machine Learning, Supervised Learning, SVM (RBF, POLY), Naive Bayes, Bayesian Networks, J48, Random Forests.

ÖZ

Makine öğrenme tabanlı spam filtreleme teknikleri analizi

NAZLI, Nazlı

Yüksek Lisans, Bilgisayar Mühendisliği Anabilim Dalı

Tez Yöneticisi: Prof. Dr Erdoğan DOĞDU

Eş - Tez Yöneticisi: Yard. Doç. Dr. Roya CHOUPANI

Şubat 2018, 79 sayfa

Bu tezde, otomatik spam eposta filtreleme problem çalışıldı. Bazı varolan makina öğrenme algoritmaları açık bir veri seti üzerinde test edildi ve sonuçlar analiz edildi. Geliştirilen metotlar makina öğrenme ve yazı sınıflandırma teknikleri kullanılarak geliştirildi. Değişik veri setleri ve test metotları karşılaştırıldı. Ağırlıklı TF-IDF, SciKit Learn tabanlı ve Word2Vec vektörizasyonu kullanarak problem çözüm için metotlar geliştirildi. Eposta yazıları için farklı vektör gösterim metotları geliştirildi ve denetimli makina öğrenme algoritmaları ile epostalar spam veya ham olarak sınıflandırıldı. WEKA yazılım aracı kullanılarak epostaların vektör gösterimleri üzerinde makina öğrenme sınıflandırma metotları uygulandı. Sınıflandırma için Destek Vektör Mekanizması SVM (POLY), SVM (RBF), Naive Bayes, Bayesian Ağları, J48 ve Rastgele Orman algoritmaları kullanıldı. Sınıflandırma yöntemlerinden elde ettiğimiz sonuçları karşılaştırdık ve analiz ettik. Sonuçlarımız Word2Vec vektörü ile SVM (Poly) algoritmasının 300 e-posta veri kümesi için 98.33% spam algılama hassasiyeti ile en iyi performansı göstermektedir.

Anahtar Kelimeler: İstenmeyen e-postaları, Makine Öğrenme, Denetimli Öğrenme, SVM (RBF,POLY), Naive Bayes, Bayesian Ağları, J48, Rasgele Ormanlar.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Prof. Dr. Erdoğan DOĞDU for his supervision, special guidance, suggestions, and encouragement through the development of this thesis. Additionally, thanks for the support of Yrd. Doç. Dr. Roya CHOUPANI during the study. I also thanks to Assoc. Prof. Reza Zare HASSANPOUR for their supports.

Special thanks go to my friends and my colleagues for their support during this study. I would also like to thank to my friend Onur GÖKER for his cooperation in related research topic. My sincere thanks also goes to my dear friends Negin BAGHERZADİ and Ali ABBASİ for supporting me writing this thesis.

Finally, I ended up doing this thesis for my father Hasan NAZLI. He was very happy that I started, he could not see what I was doing. It is a pleasure to express my special thanks to my family for their valuable support. Thank you very much for my mother Yıldız NAZLI, my sister Sümeyra NAZLI and my brother Bilal Murat NAZLI for all their support, help and beliefs.

TABLE OF CONTENTS

STATEMENT OF NON-PLAGIARISM PAGE	iii
ABSTRACT	iv
ÖZ	v
ACKNOWLEDGEMENTS	vi
TABLE OF CONTENTS	vii
LIST OF TABLES	ix
LIST OF FIGURES	xi
LIST OF ABBREVIATIONS	xiii
CHAPTER 1	1
INTRODUCTION	1
1.1 Background	2
1.2 Problem Statement	3
1.3 Contributions	5
1.4 Thesis Organization	6
CHAPTER 2	7
RELATED WORK	7
2.1 Machine Learning Methods	8
2.2 Spam Detection Algorithms	10
CHAPTER 3	13
MACHINE LEARNING-BASED SPAM FILTERING	13
3.1 Data Representation	13
3.1.1 Weighted TF-IDF Vectorization	14
3.1.2 Gradient Boosting and TF-IDF using SciKit Learn	14

3.1.3 Word2Vec	15
3.2 Machine Learning Based Classification.....	15
3.2.1 Naive Bayes	16
3.2.2 Bayesian Network	16
3.2.3 Support Vector Machine (SVM).....	16
3.2.4 J48	17
3.2.5 Random Forest	17
CHAPTER 4	18
EVALUATION.....	18
4.1 Dataset.....	18
4.2 Tools and Libraries	20
4.3 Validation metrics	21
4.3.1 Weighted TF-IDF.....	22
4.3.2 SciKit Learn with TF-IDF.....	39
4.3.3 Word2Vec	39
4.4 Test Results	56
CHAPTER 5	57
CONCLUSION	57
REFERENCES.....	58
APPENDICES A.....	63
CURRICULUM VITAE	63

LIST OF TABLES

1. Table 1-Spam Detection Literature Taxonomy	8
2. Table 2-01-31 and 32-58 attribute accuracy and f-measure.....	11
3. Table 3-6 folds and 10 folds SVM, Random Forest and AdaBoost Algorithms Accuracy	11
4. Table 4-Vector names, generation methods and their sizes.....	20
5. Table 5-F-Measure to TP-FP-FN-TN	22
6. Table 6-Accuracy Results for Vector Representation based on Weighing TF-IDF	23
7. Table 7-The Results for Weighting F-Measures to 300&500 data sets between NB & BN	27
8. Table 8-The Results for Weighting F-Measures to 300&500 data sets between SVM (RBF & POLY).....	28
9. Table 9-The Results for Weighting F-Measures to 300&500 data sets between J48 & RF.....	28
10. Table 10-The Results for Weighting F-Measures to 1000&2000 data sets between NB & BN	30
11. Table 11-The Results for Weighting F-Measures to 1000&2000 data sets between SVM (RBF & POLY).....	31
12. Table 12-The Results for Weighting F-Measures to 1000&2000 data sets between J48 & RF.....	32
13. Table 13-The Results for Weighting F-Measures to 5000&1000 data sets between NB & BN	33
14. Table 14-The Results for Weighting F-Measures to 5000&10000 data sets between SVM (RBF & POLY).....	34

15. Table 15-The Results for Weighting F-Measures to 5000&10000 data sets between J48 & RF.....	35
16. Table 16-SciKit Learn Tools Results.....	39
17. Table 17-Test Results for Word2Vec Accuracy	40
18. Table 18-The Results for Word2Vec F-Measures to 300&500 data sets between NB & BN	44
19. Table 19-The Results for Word2Vec F-Measures to 300&500 data sets between SVM (RBF&POLY).....	45
20. Table 20-The Results for Word2Vec F-Measures to 300&500 data sets between J48 & RF.....	46
21. Table 21-The Results for Word2Vec F-Measures to 1000&2000 data sets between NB & BN	47
22. Table 22-The Results for Word2Vec F-Measures to 1000&2000 data sets between SVM (RBF&POLY).....	48
23. Table 23-The Results for Word2Vec F-Measures to 1000&2000 data sets between J48 & RF.....	49
24. Table 24-The Results for Word2Vec F-Measures to 5000&10000 data sets between NB & BN	50
25. Table 25-The Results for Word2Vec F-Measures to 5000&10000 data sets between SVM (RBF & POLY).....	51
26. Table 26-The Results for Word2Vec F-Measures to 5000&10000 data sets between J48 & RF.....	52

LIST OF FIGURES

1. Figure 1-Life cycle of phishing email [9]	4
2. Figure 2-Kernel Machine [44]	17
3. Figure 3-Accuracy comparison to 300 ham and 300 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms	24
4. Figure 4-Accuracy comparison to 500 ham and 500 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms	24
5. Figure 5-Accuracy comparison to 1000 ham and 1000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms	25
6. Figure 6-Accuracy comparison to 2000 ham and 2000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms	25
7. Figure 7-Accuracy comparison to 5000 ham and 5000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms	26
8. Figure 8-Accuracy comparison to 10000 ham and 10000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms	26
9. Figure 9-F-Measure comparison to 300 ham and 300 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms	36
10. Figure 10-F-Measure comparison to 500 ham and 500 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms	36
11. Figure 11-F-Measure comparison to 1000 ham and 1000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms	37
12. Figure 12-F-Measure comparison to 2000 ham and 2000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms	37
13. Figure 13- F-Measure comparison to 5000 ham and 5000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms	38

14. Figure 14-F-Measure comparison to 10000 ham and 10000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms	38
15. Figure 15-Different data sets and results for SciKit Learn Tools	39
16. Figure 16-Accuracy comparison to 300 ham and 300 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms	41
17. Figure 17-Accuracy comparison to 500 ham and 500 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms	41
18. Figure 18-Accuracy comparison to 1000 ham and 1000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms	42
19. Figure 19-Accuracy comparison to 2000 ham and 2000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms	42
20. Figure 20-Accuracy comparison to 5000 ham and 5000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms	43
21. Figure 21-Accuracy comparison to 10000 ham and 10000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms	43
22. Figure 22- F-Measure comparison to 300 ham and 300 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms	53
23. Figure 23-F-Measure comparison to 500 ham and 500 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms	53
24. Figure 24-F-Measure comparison to 1000 ham and 1000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms	54
25. Figure 25-F-Measure comparison to 2000 ham and 2000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms	54
26. Figure 26-F-Measure comparison to 5000 ham and 5000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms	55
27. Figure 27-F-Measure comparison to 10000 ham and 10000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms.....	55

LIST OF ABBREVIATIONS

SVM	Support Vector Machine
LIBSVM	A Library for Support Vector Machines
SMO	Sequential Minimal Optimizations
APWG	Anti-Phishing Work Group
TF-IDF	Term Frequency-Inverse Document Frequency
BN	Bayesian Network
NB	Naive Bayes'
RBF	Radial Basis Function
POLY	Polynomial
ML	Machine Learning
SL	Supervised Learning
USL	Unsupervised Learning
SSL	Semi-Supervised Learning
NLTK	Natural Language Toolkit
TP	True Positive
FN	False Negative
TN	True Negative
FP	False Positive
URL	Uniform Resource Locator

CHAPTER 1

INTRODUCTION

Worldwide e-mail usage in 2015 is estimated to be about 2.6 billion, and in 2019 it is expected to be over 2.9 billion. The number of e-mails sent and received daily in 2015 is over 205 million. These figures show an average increase of 3% per annum and this study informs that 246 million mail will provide daily flow until the end of 2019 [1]. The primary aim of this study is developing a spam filtering technique based on machine learning analysis. Our research has shown that the increase in the use of the Internet is directly proportional to the increase in the infiltration of malware into systems. Spam emails, which are also called phishing emails, have become increasingly common. It is difficult to detect or prevent phishing attacks. Even the experienced or careful users in this regard may be the subject of these attacks. Attackers copy their known pages or make a few changes in their own way to make unreal web pages and emails. In the United States, online payment systems are quite common and have become widespread in spam (phishing) e-mails. In June 2001 E-Gold attack with no measurement was carried out. In the last quarter of 2003, hundreds of addresses were recorded in the appearance of legitimate sites such as phishing eBay and PayPal. These fake sites, which escape the attention of users, are very similar to their realities. In fact, PayPal or eBay has used a fake email worm program to access their accounts. As a result, site users, who were victims, were seized by this system under the name of updating all their information by being directed to a fake site [2]. Many companies in the US lose millions of dollars a year due to cybercrime. Cybercrime puts in danger of more than 130 million user accounts, and many companies such as PayPal and eBay suffered from it [3]. Users cannot tell in general easily the difference between real or attacker-organized emails.

Attackers are making use of many different kinds of fraud methods since the beginning of internet and the Web. Measures can be taken against these attacks. There are many ways to prevent these attacks.

In this thesis, we have examined some of the methods implemented by machine learning algorithms on spam detection and filtering. Machine learning methods are divided into supervised learning (SL), unsupervised learning (USL) and semi-supervised learning (SSL). First, in “Supervised Learning” all data are labeled and the method undergoes a learning process from the training data set. The algorithm learns to guess its output from the input data. Second, in “Unsupervised Learning” the data is unlabeled, and it is learned from the input data by using the algorithm’s modeling method. Thirdly, in “Semi-supervised Learning” some data are labeled, but most of them are unlabeled since labeling is expensive and time consuming. So this method uses a mixture of supervised and unsupervised techniques. [4].

In this thesis, we studied and used supervised learning methods. There are many different algorithms in this approach. Some of the most popular and successful ones are for example, Support Vector Machine (SVM), Naive Bayes, and J48. We used some of these methods in our tests. And, we obtained different results with different methods and test datasets.

1.1 Background

This section focuses on identifying the research area and scope. Nowadays, the measures taken with the increase of the cyber-attacks are growing. Many fraud schemes can easily acquire personal information and credit card information over the Internet. In a work done in 2011, statistics show that 40% of all emails are spam. Spam traffic care reaches about 15.4 billion e-mails a day and can cost internet users \$355 million a year in costs related to damages and loss of work [5].

On the other hand, the Statista Global Consumer Survey investigated the interaction of consumer behavior with products. In a study conducted by Kaspersky Lab in March of 2016, more than 22 million malicious spam was detected [6]. There are many systems developed to prevent this.

We aim to prevent spam emails by developing a detection system, using machine learning (ML) algorithms. We compared and analyzed the results from different ML algorithms that are commonly used in this area.

A detailed description of our methods is presented in the remaining sections. For better analysis, we discuss the methods for analyzing the emails first. Our approach is to analyze the text in emails only, instead of all other data such as images, links, attachments, etc. since they require much more complex learning methods. We therefore chose a simple method for email classification.

We also concentrated on email representation as digital data in the form of feature vectors. We tried three different vector representation methods.

1.2 Problem Statement

It is known that, the APWG Cryptocurrency Working Group regularly publishes about the attacks. It has a structure updated by MarkMonitor. According to APWG, in the 2nd Quarter of 2017 the number of unique phishing websites detected has been increasing day by day [7]. In the 3rd Quarter of 2017, it was the trend report shown us, this increase continues and the number of detected phishing emails during the period was 190,942. Although the number of detected sites is lower than the previous one, the phishing reports for the second quarter are 273,395 while the number of phishing for the third quarter is 296,208. In fact, 54,631 different sites were detected, but HTTPS protected four of them [8]. In our study, we used machine learning techniques to detect spam emails. We used some data sets containing clean (ham) e-mails and phishing (spam) emails. One of our data sets includes 10k ham and spam email samples. These attacks aim to steal the passwords of users, social security numbers of account numbers, or personal information. Attackers send e-mails to users in various ways. Figure 1, shows a scenario of the phishing email life cycle where an email is sent to an email receiver and that a user clicks on a phishing URL via this post and that the web site is displayed and then the victim enters the information into the system [9].

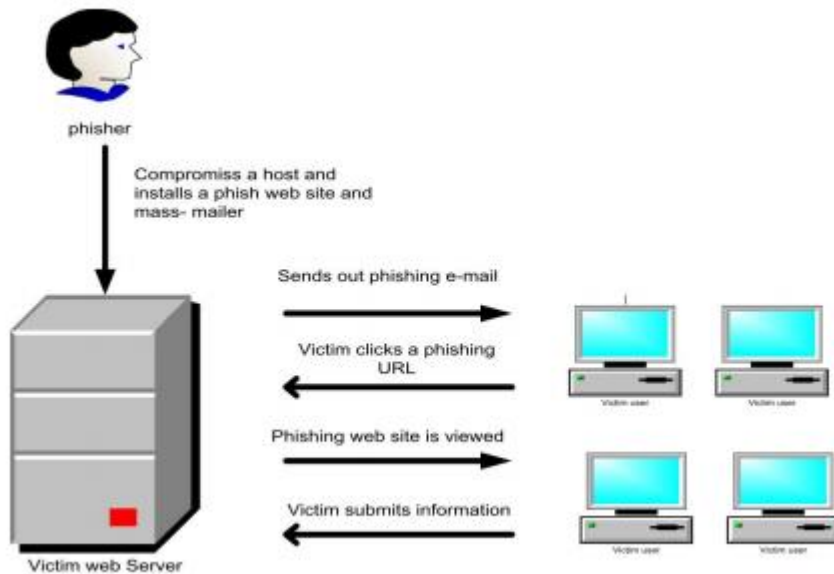


Figure 1-Life cycle of phishing email [9]

The stolen information may belong to reputable financial institutions, business associations or known organizations. The organizers of these attacks work systematically and first collect the email addresses of the users. Then they attack these addresses with fake accounts. These spam mails are spreading very quickly, and an urgent precaution should be taken when this danger is taken into consideration. This measure must be fast and robust.

Non machine learning techniques divides mails into three parts: blacklist and whitelist, signatures and mail header checking [10]. But many methods are costly and take a long time. There are often studies in which content-based approaches are made. When we think of billions of email traffic on the internet, we need to make a faster and more robust classification to catch phishing mails. We need to perform the process in a limited time to analyze the results of this classification process. So we investigated the fast algorithms needed to solve this problem in our research.

We have achieved some success with some sorting methods that we have identified. The methods used for SVM (Classification) are very effective and distinctive. The Machine Learning techniques used here are from three classes: AdaBoost, Naive Bayes and Support Vector Machines. We especially considered SVM classifier in this study. This classification was studied as supervised learning and studied in two groups. SVM plays an important role in kernel selection.

SVM is the most important algorithm of data mining. It gives more successful results than many techniques. So we preferred this method with single variable variance analysis. The most successful kernel function has been proved to be a RBG radial-based function and other one is POLY polynomial function. Bayes network is a learning model. In other words, each node of a row tree uses the child nodes of the parent row. Naive Bayes is the same of Bayesian probability function. These algorithms are WEKA's classification algorithm. Weka is implemented in java using j48 C4.5. The last method is Random Forest algorithm. Spam and ham mails from the Enron datasets was arranged and separated, and added to the system in separate files.

The implementation is done using Python language. Then, the words in emails were parsed and the words in the content were separated and the development made on them. Firstly, we used the document frequency TF-IDF (Term Frequency-Inverse Document Frequency) method to calculate the weight, and obtained vectors as much as the data sets used [11]. These data sets were then processed with SciKit-learn. Success rates obtained were added to the table. SciKit-learn is a machine learning software used with Python programming languages [12]. Support vector machines have various classification, regression and clustering algorithms such as random forests, k-means. NumPy and SciPy work with numerical libraries. Finally, the google Word2Vec library was used [13]. It includes libraries such as Gensim, NumPy. Vector creation process was done with Gensim. The results obtained were then processed using Weka.

1.3 Contributions

In this thesis, our contributions are as follows:

- An open dataset, Enron email dataset, is parsed (for body and subject) and sampled into a number of datasets of varying sizes with equal spam and clean email counts for testing different ML algorithms for spam detection.
- Three different vector representation methods for email text are performed on the test datasets.
- Different ML algorithms are then applied on vector representations of email text in all test datasets, and the results are analyzed and reported in this thesis.

1.4 Thesis Organization

This thesis is divided into five chapters. The chapters cover all researches that we conducted to obtain an analysis of machine learning-based spam filtering techniques, and all the results obtained are examined as described below.

Chapter 1 is an introduction and background for Spam filtering based on machine learning and objectives of this thesis. This chapter also includes problem statement and our contributions.

Chapter 2 explains the related work for spam detection algorithms and machine learning methods.

Chapter 3 presents the data representation methods we developed and used for email text. We also present the method for email classification.

In Chapter 4, the experiments of our methods in relation with the data sets, tools and results are presented. We also present the results of our test and analyze them.

Chapter 5 concludes and presents the future work in this area.

CHAPTER 2

RELATED WORK

In this part of the thesis, the other studies are reviewed on the analysis of spam filtering based on machine learning algorithms. We have studied the work done in this section depicting our position under the name of two titles. These titles are related to machine learning methods and spam detection algorithms. This part helped us find answers to the questions: What methods better suit our application? And, how can we get better results?

In this research, we mostly used Google Scholar and searched for articles and theses. We have searched for "Spam emails filter", "Spam e-mail detection", "Spam e-mail classification and analysis with machine learning", "Analysis of machine learning-based spam filtering techniques" as search keywords. We made choices by specifying a working method. We received articles from the universities' electronic libraries.

To answer the above questions, we have created a taxonomy table (Table 1). In the research that we made, we classified the studies in the literature. The methodology of the studies done, the algorithms used, the data sets used, and the success rates.

Table 1-Spam Detection Literature Taxonomy

Algorithm	Data Sets	Highest Success Rate	References
Bayesian Net	CSDMC2010, Spam Assassin, and LingSpam,1171 raw phishing emails and 1718 legitimate emails	85.45%	[14], [15]
Naive Bayes	Discretized , RUL:6000 emails with the spam rate 37.04%	99.46%	[11], [14], [16], [17], [18], [19], [20],
SVM	1171 raw phishing emails and 1718 legitimate emails, Discretized , RUL:6000 emails with the spam rate 37.04%	96.90%	[11], [14], [15], [16], [17], [20], [21], [22], [23], [24], [25], [26], , [27],
J48	4601 messages:1813 (%39) by Hopkins et al as spam, others are Legal messages by Forman.	92.6 %	[28], [29]
Random Forest	4601 messages:1813 (%39) by Hopkins et al as spam, others are Legal messages by Forman.	93.75%	[16],[28], [30]

2.1 Machine Learning Methods

There are some research works that applied machine learning methods to spam email detection. In the following we provide a brief description of a few spam filtering techniques.

In [17], the authors proposed a spam detection algorithm based on Machine Learning approach. They used the Cumulative Weighted Remainder (CWS) concept because they wanted to attain a higher rate in detecting spam mails, and the method could detect most of the unwanted mails. They provide accurate and dynamic filtering for emails.

In addition, it uses User-based Learning Algorithm (UMLA) based classifiers and Support Vector Machines method, Naive-Bayesian, and Decision Tree algorithms. There are data sets with 1000 email used in their research.

These data sets play an important role in improving spam filtering performance, obtained by testing techniques on data clusters with different criteria. The results are 88% false positive, false positive 8%, false positive compared to the selected data set. 8%, and true negative is 97.3%. The accuracy of the algorithm based on the matrix technique is 95% of the total accuracy.

In [20], the authors proposed the application of the most popular machine learning methods (Bayesian classification, k-NN, YSA, SVMs, artificial immune system and Rough sets) for spam e-mail classification. The methods they use are: Artificial Neural Networks classifier, K-nearest neighbor classifier method, Naïve Bayes classifier, Support Vector Machines classifier, Artificial Immune System classifier, and Rough sets classifier. They explain all of use algorithm in their work. The results of their works:

- Ham mails train set: 2378, test set: 1400
- Spam mails train set: 1398, test set: 824
- 100 features are selected to measure the performance of 6 algorithms.
- The evaluation metrics are Spam Recall, Spam Precision and Accuracy.
- Best solution Spam Recall values: 98, 46% of Naive Bayes algorithm.
- Best solution Spam Precision values: 99, 66% of Naive Bayes algorithm.
- Best solution for Accuracy values: 99, 46% of Naive Bayes algorithm & 97, 42% of Root sets.

In this study, Naive Bayes and Root sets have yielded a more accurate and successful result. Therefore, they announced that hybrid systems have created a more successful spam filter.

In [25], the authors reported that a different kernel functions (linear, polynomial, RBF, sigmoid) have been implemented in the spam data set in one of the support vector machine (SVM) methods with different parameters as machine learning algorithm. The test, applied using different parameters, compares the SVM performance of all cores (linear, polynomial, RBF, sigmoid). We evaluated the spam-based dataset to get good results. The results are shown in the table below. The results obtained with different train and test clusters using Iris data set with 3 classes, pen digit dataset with 10 classes, and News20 dataset with 20 classes. Their performance tested using LIBSVM tool with different SVM kernels: In linear kernel 50% training set and 50% test set accuracy is 92.4381%. In polynomial kernel 40% training set and 60% test set accuracy is 78.3050%. In RBF kernel 90% training set and 10% test set accuracy is 87.2817%. In sigmoid kernel 90% training set and 10% test set accuracy is 66.5944%.

2.2 Spam Detection Algorithms

There are quite a few studies on spam detection topic. In the following we provide a brief description of a few spam filtering techniques.

In [28], the authors examined tree algorithms, and methods and attribute optimizations related to data mining. Some of them are: Random Tree Algorithm, Extra Tree Algorithm, ADTree Algorithm, J48 Tree Algorithm, NBTree Algorithm and Random Forest Tree Algorithm. These are described in detail in the work and the applied process is added as a result with 58 features.

In the work done with 58 attributes the results are:

- Random tree: 90,65%
- ADTree: 91,55%
- J48: 92,6%
- Random Forest: 93,75%
- Extra Tree: 90,45%
- Simple Chart: 91,9%

Particle Swarm Optimization (PSO) and attributes 01-31 and 32-58.

Table 2-01-31 and 32-58 attribute accuracy and f-measure

Attribute	Accuracy	F-Measure
01-31	91,45	91,436
32-58	76,50,	76,388

Then the authors applied Gain Ratio algorithm. They conclude that Random Forest algorithm is a better classification than the other algorithms.

In [30], the authors proposed an approach to improve the performance of SVM using different techniques. Support Vector Machines (SVMs) and Decision Trees (DTs) have performed very well in identifying spam emails and have stated that some features for classifiers such as SVMs, AdaBoost and Random Forests (RF) can also affect direct and F-Score performances. The data set they use is Enron data set. Feature elements are divided and processed into different sizes of 50, 100, 150, 200, 250 and 300 for estimation. All characters are processed to stop word and stemming. Their results are firstly related to SVM, RF and AdaBoots with 6 folds cross-validation. Then, 10 folds cross validation. 6 folds Bag of Words (BoW) and Information Gain (IG) have values ("1" or "0") are used for joint display. Using the RBF kernel function, with $C = 10$ and $\lambda = 0.1$. 10 folds SVM using the RBF kernel function, with the parameters $C = 10$ and $\lambda = 0.1$.

Table 3-6 folds and 10 folds SVM, Random Forest and AdaBoost Algorithms Accuracy

Folds \ Accuracy	SVM (%)	Random Forest (%)	AdaBoots (%)
6 Folds	89,02	95,11	93,57
10 Folds	97,89	97,94	97,03

In [31], the authors proposed an intelligent model for detection of phishing emails that is doing a feature separation with different email fragments that are bound to a preprocessing phase. Extracted features are classified using the J48 classification and are used with a total of 23 properties. Then ten times the cross validity is applied for training and testing.

It is claimed to be the best algorithm that can be used and the results are showing this model has 98.87% accuracy. In addition, the most commonly used random forest algorithm is using a tried and tested data set. This is where the success of the ten different classification algorithms is compared.

In [32], the authors have made a research on performing well in perception of phishing. In the study, 600 subjects were used to collect data where the survey was conducted. For each participant, a total of 16 emails were randomly selected from 50 phishing e-mails and original business e-mails. This helps both in behavioral decision making and in the advancement of theories in phonemic perception. Social cognitive theory, behavioral theories, motivation theory is a topic in which researchers continue to discuss judicial bias. Behavioral decision-making research, or simply manipulation of task difficulty, leads to over-reliance.

CHAPTER 3

MACHINE LEARNING-BASED SPAM FILTERING

The purpose of this chapter is to explain two machine learning based spam filtering methods. It is difficult to process and analyze very large amounts of data. Our goal is to increase the success rate in testing by learning with train data sets. Machine learning methods have been developed for this purpose. We will explain the algorithms and methods we use in our work in this chapter. Analysis of the data in the process of data transfer is done with data mining and some applications are available nowadays. We can list them as handwriting / book reading, e-commerce, evaluating credit requests, gene micro analyses (eye reading, fingerprinting, face and voice analysis). We will first explain how to create three vectors that we have used. These are Weighted TF-IDF, SciKit Learn based, and Word2vec vector, respectively. We will then explain and analysis the algorithms we use to get the thesis results. Naive Bayes, Bayesian Network, SVM (RBF and Poly) kernel, J48 and Random Forest algorithms respectively.

3.1 Data Representation

In order to classify the spam traits, we had to translate the words in it into a numerical value. Thus, we could make a classification and similarity finding process using machine learning algorithms. In the literature research, vector creation process was applied to categorize the texts seen. This vector numerically represents the corresponding mails. We have developed a method using vectors and we have studied them in this thesis. We performed our work on the text parts of the emails. First we calculated the TF-IDF weight with the frequencies of the email terms.

Then we created a vector with TF-IDF, gradient boosting and the SciKit Learn library. Finally we used Word2Vec to get a vector. In this section the methods, algorithms and detailed explanations of our test methods have been given.

3.1.1 Weighted TF-IDF Vectorization

TF-IDF is a weighted factor that is a statistical calculation of the importance of a term in a document. We used this method to classify the words in the text. Stop Words can be used in filtering. Stop word is frequently used conjunctions, punctuation marks, lexical words and functional words. It means words without terms [33].

Term Frequency is used to calculate the weight of a term in a document. Inverse Document Frequency is used to find the number of repetition in more than one document, and to know if this is a term, such as "Stop Words". For this, the absolute value of the TF-IDF logarithm is taken.

$$TF-IDF = TF \times IDF$$

We did this calculation for ham and spam emails. We created a vector by taking a weight difference. Then calculate the wait for just a word in these documents.

Our Weighting formula:

$$W_i = W_{hi} - W_{pi}$$

In this formula:

W_i : calculate weight value.

W_{hi} : ham emails weighting.

W_{pi} : phishing emails weighting.

3.1.2 Gradient Boosting and TF-IDF using SciKit Learn

We use SciKit-Learn to create vectors. SciKit-learn are a Python module used with machine learning algorithms for supervised and unsupervised problems [12].

This program focuses on providing information to people who are not experts using the intended high level language of learning. It is based on ease of use, performance and documentation and requires the use of NumPy and SciPy. So it is a simplified academic as well as commercial tool.

Gradient boosting (XGBoost) is composed of extreme libraries. It is a library that allows multiple algorithms to work together better [34]. The main idea is developing a strong classifier by combining a set of weak classifiers. The assumption is the problem is a binary class and an odd number of weak classifiers are available. The problem is presented to these classifiers and the result is obtained by voting over the results of these weak classifiers.

3.1.3 Word2Vec

The skip-gram model learns a large number of high quality distributed vector presentations. Vectors have several extensions that improve both quality and training. With sampling of frequently used words, we can achieve a significant acceleration [35].

We can also learn more regular vocabulary. Tomas Mikolov and his colleagues say that it is possible to learn good vector presentations for millions of phrases [36].

We use their pre-trained vectors in Google News data sets to train (about 100 billion words). This model contains 300-dimensional vector for 3 billion words and phrases. These expressions were obtained using a simple data approach. We use archive data Google News including 1.5 GB data [37].

We got an error when we started our code. The reason is that Gensim allocates a large matrix to hold all word vectors, and python is 32 bits wide. This is calculating that:

$3 \text{ million words} * 300 \text{ features} * 4 \text{ bytes / feature} = \sim 3.35 \text{ GB}$ [37].

3.2 Machine Learning Based Classification

In this section we present the machine learning algorithms we used for email classification. We explain all algorithms in this part in the following order: Naive Bayes, Bayesian Network, Support Vector Machine algorithms of RBF and POLY kernel, J48 and Random Forest algorithms.

3.2.1 Naive Bayes

Naive Bayes name comes from the famous mathematician Thomas Bayes. The algorithm uses the Bayes theorem. It is a classification algorithm. In ML the algorithm refers to the same family of probabilistic theorem. It assumes that all attributes are independent when the value of the class variable is considered. This classification is based on the simple probability principle. It contains frequency and data set value [38].

3.2.2 Bayesian Network

Bayesian Network is based on probability function and is an efficient algorithm for learning. This algorithm's learning is based on constraints and perform the join process effectively by exploiting the search-point technique [39]. It solves decision problems under call influence diagrams. These algorithms search all data sets. Since the search area is very large, the Bayesian can pass this search by examining the candidates, which are usually illogical [40]. We used this algorithm and added the results we got into the thesis.

3.2.3 Support Vector Machine (SVM)

Vapnik (1995) has developed the foundation of Support Vector Machines (SVM). It is gaining popularity due to many attractive features and better performance [41].

We examined the SVM algorithm under two method headings, which are RBF and Poly kernel. SVM is very effective and distinctive algorithms to classify our data sets. Sometimes RBG radial-based function linear classifications may not be possible in the data set. Each data is mapped to the upper property space and this new space uses a hyperplane classification method. While the RBF kernel is used for linear non-separable data sets and other one is POLY kernel polynomial function is used to determine the hyperplane of the non-linearly disjoint data set [42]. We focus on our classifications and conclude that these algorithms are better than others.

We calculate SVM accuracy and f-measure using WEKA tools and SMO (Sequential Minimal Optimization) function .SMO is a fast algorithm for partitioning data into multiple clusters using Support Vector. It solves the problem repeatedly for each subset. SMO, also found in WEKA, is similar to SVM [43].

Figure 2 shows SVM Polynomial Kernel and SVM RBF Kernel graphic.

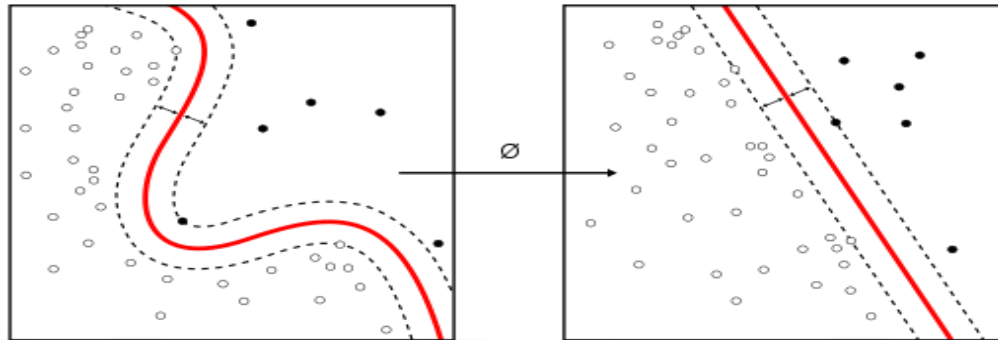


Figure 2-Kernel Machine [44]

3.2.4 J48

J48 is a tree algorithm and implementation of C4.5 decision tree algorithm [45] in the WEKA data mining tools. Classification steps of j48 are simple and fast. According to the researches, it is generally an algorithm used to compare the data sets against the naive Bayes algorithm.

3.2.5 Random Forest

The decision tree learning method is one of machine learning topics. Random Forest algorithm is a supervised classification algorithm. It is also understood from the name that to form a random forest it is necessary to obtain a result depending on the number of trees in the forest. Decision trees perform better than algorithms such as support vector machines and naïve Bayes. We can say that the random forest is a good choice for these tasks and that it works fast in large, high-dimensional databases [46]. The more trees we have, the more we will learn and obtain a more accurate result. This is the algorithm we used with WEKA default value.

CHAPTER 4

EVALUATION

We used Python programming language for development. Python has a rich set of libraries useful for many tasks. We have used the following Python libraries: NLTK, Word2Vec, Gensim, NumPy, and SciKit. Gensim is a python implementation of Word2Vec method. Using these libraries we created the vectors of for email datasets we have chosen. We also used Weka tool to test ML algorithms on the classification of emails as spam or ham. Weka has an implementation of almost all ML algorithms including the ones we have chosen: SVM (poly), SVM (RBF), Naive Bayes, Bayesian Networks, J48 and Random Forest algorithms. We compare the success of the algorithms. We used all the vectors we produced as input. We also used the default values of the algorithms for testing. The results are based on observations, and the dataset labels. We report below.

First we explain the datasets we have used. Then, the tools used are explained. And, we report and analyze the results at the end of this chapter.

4.1 Dataset

In earlier test we used an open email phishing dates from Comodo Inc. But this dataset is limited with only 100 phishing emails and no cleans emails. Therefore we used another open but large dataset from the bankrupted Enron company¹. This data set has labeled emails; therefore it is very useful.

¹ Enron Dataset of Carnegie Mellon University, School of Computer Science
<https://www.cs.cmu.edu/~enron/>

Spam emails, as we mentioned earlier, are mostly advertisement emails that are sent to users mostly without their consent, which may be able to harm the users via phishing or other ways. Ham, or clean, e-mails are users' correspondence in digital media, electronic correspondence. Inside ham emails can be ads that do not have anything fake to steal your information. We used all the e-mails and parsed the words in the body of the mails and made them into vectors using tools and also our implementations. We tried 3 different vector representations by applying different vectorization methods on email text.

We first constructed the following subsamples of the Enron dataset:

- 300 ham email + 300 spam email text
- 500 ham email + 500 spam email text
- 1000 ham email + 1000 spam email text
- 2000 ham email + 2000 spam email text
- 5000 ham email + 5000 spam email text
- 10000 ham email + 10000 spam email text

Table 4 lists the spam and ham emails word vectors files and their sizes using different production methods.

Table 4-Vector names, generation methods and their sizes.

VECTOR	METHOD	SIZE (MB)
300 ham + 300 spam	Word2Vec	1.3
500 ham + 500 spam	Word2Vec	2.2
1k ham + 1k spam	Word2Vec	4.4
2k ham + 2k spam	Word2Vec	8.9
5k ham + 5k spam	Word2Vec	22.2
10k ham + 10k spam	Word2Vec	44.4
300 ham + 300 spam	Weighted TF-IDF	0.9
500 ham + 500 spam	Weighted TF-IDF	1.5
1k ham + 1k spam	Weighted TF-IDF	3
2k ham + 2k spam	Weighted TF-IDF	6
5k ham + 5k spam	Weighted TF-IDF	15
10k ham + 10k spam	Weighted TF-IDF	30

4.2 Tools and Libraries

The programming language we used is python 3.5 and python 2.7. We used Pycharm 2017.3.3 and Visual Studio Code 1.20.1 for development for this language. We used Python libraries NumPy, SciKit-Learn², Gensim, Word2Vec and NLTK³ for vector generation.

For classification ML algorithms testing we used WEKA tool. WEKA 3-8-2 stable version and WEKA 3-9-2 developer version are used. WEKA⁴ is simple to use with a graphical user interface (GUI). WEKA stands for Waikato Environment for Knowledge Analysis. Weka tools helped us for in testing ML algorithms we have chosen, such as SVM, Naive Bayes, J48, ext.

² SciKit Learn

<http://scikit-learn.org/>

³Natural Languages Toolkit

<https://www.nltk.org/>

⁴WEKA-University of Waikato

<https://www.cs.waikato.ac.nz/ml/weka/downloading.html>

Our vectors we generated have 300 features and (1/0) for label (as spam or clean). The vectors files are input to Weka in CSV format.

We used three different test options to calculate the accuracy and f-measures, as provided by Weka tool for validation. These are:

- **Percentage Split 66%:** 66% of the dataset is used for training the model and 34% is used for the testing (validation). Once the system is trained with the training cluster, the test part is used on the trained model to measure the success.
- **Percentage Split 80%:** 80% of the dataset is used for training and 20% for testing.
- **K-Fold Cross-Validation:** To remove the bias in training/test part selection, this method is widely used in almost all ML tools. The most preferred k value in the literature is 10. In k-fold cross validation, the dataset is divided into k equal parts. If we select k=10, our data set will be divided into 10 equal parts. In each fold, one part selected for testing, the rest of the parts are used for training the model and the test part is tested on the model. At the end, the averages of accuracies from all folds are calculated as the final result.

4.3 Validation metrics

We calculate the accuracy, which is percentage of correct predictions in each test.

And accuracy is calculated with the formula:

$$Accuracy = \frac{|TP| + |TN|}{|TP| + |TN| + |FP| + |FN|}$$

where,

- True Positive (TP): a spam email is correctly predicted as spam.
- True Negative (TN): a ham email is correctly predicted as ham.
- False positive (FP): a ham email is incorrectly predicted as spam.
- False Negative (FN): a spam email is incorrectly predicted as ham.

F-measure is used to measure the validation success mostly in the literature. Precision and recall values are needed for this [7].

Precision formula is:

$$Precision = \frac{|TP|}{|TP| + |FP|}$$

Recall formula is:

$$Recall = \frac{|TP|}{|TP| + |FN|}$$

F-Measure formula is then:

$$F - Measure = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Table 5 shows how the TP, FP, FN and TN values are used. In this table, we can observe how we place the values of the test results of the SPAM and HAM mails, and the values used to calculate the f-measure. This table shows the semantic relation between spam and clean mail in the c / d section with f-measure value.

Table 5-F-Measure to TP-FP-FN-TN

SPAM	HAM		c/d
TP	FP	SPAM	
FN	TN	HAM	

c/d: correct classified/ total number

4.3.1 Weighted TF-IDF

In the data set we use, train and test data set distributions are as shown in the following tables. The data sets used are divided into 300, 500, 1000, 2000, 5000 and 10.000. We applied these algorithms in three different test, 10 folds (average of 10 train / test data set), 66% and 80%, and we obtained different results. The best accuracy score for each data set is highlighted in Table 6.

Table 6-Accuracy Results for Vector Representation based on WeighingTF-IDF

Data Sets	Process	Naive Bayes (%)	Bayesian Network (%)	SVM (RBF)(%)	SVM (POLY) (%)	J48 (%)	Random Forest (%)
300 + 300	10 Folds	62	51.83	58	61.66	54.66	68
	%66 Percentage	61.76	51.47	55.39	65.68	59.31	71.07
	%80 Percentage	65	48.33	46.66	67.5	60.83	73.33
500 + 500	10 Folds	59.9	55.5	53.6	51.5	54.2	63.8
	%66 Percentage	61.47	56.17	46.17	53.23	56.17	58.82
	%80 Percentage	63.5	46.5	44.5	49	58	53.5
1000 + 1000	10 Folds	53.1	57.15	52.7	56.45	55.6	59
	%66 Percentage	54.55	53.82	52.5	55.58	52.79	56.76
	%80 Percentage	52.5	54.25	51.75	54	54	58
2000 + 2000	10 Folds	53.45	52.85	53.4	53.4	52.15	55.25
	%66 Percentage	54.92	51.76	53.01	53.97	51.91	54.7
	%80 Percentage	52.25	50.87	48.5	53.12	50.87	57.37
5000+ 5000	10 Folds	53.12	52.75	51.18	53.02	52.54	53.65
	%66 Percentage	53.17	52.64	52.5	53.7	52.14	54.35
	%80 Percentage	53.45	52.6	51.1	53.7	52.6	54
10000 + 10000	10 Folds	50.38	52.39	50.75	52.36	52.39	53.39
	%66 Percentage	50.75	52.22	50.69	52.61	52.25	53.08
	%80 Percentage	50.05	52.92	49.52	53.52	52.9	53.6

Figure 3 shows the results in a bar graph for 300+300 dataset. For 10-fold test, the best algorithm is Random Forest (RF), for 66% test RF performs the best, and for 80% test again RF performs the best.

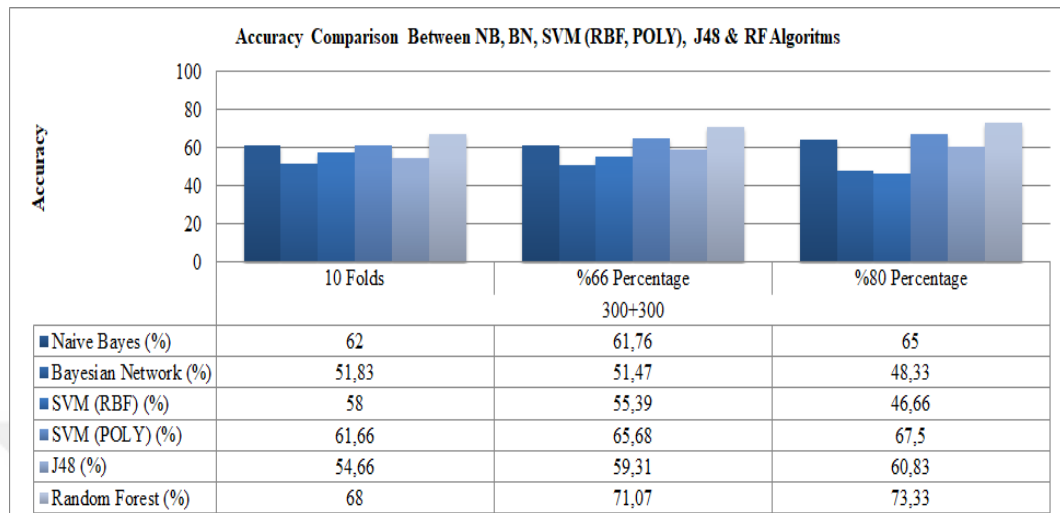


Figure 3-Accuracy comparison to 300 ham and 300 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms

As seen on Figure 4 for 500+500 dataset, in 10-fold test method, the best result is obtained by using Random Forest algorithm. For 66% method, the best score is obtained by Naïve Bayes and for 80% method; the best score is obtained by Naïve Bayes again. Forest with 300 ham + 300 spam data.

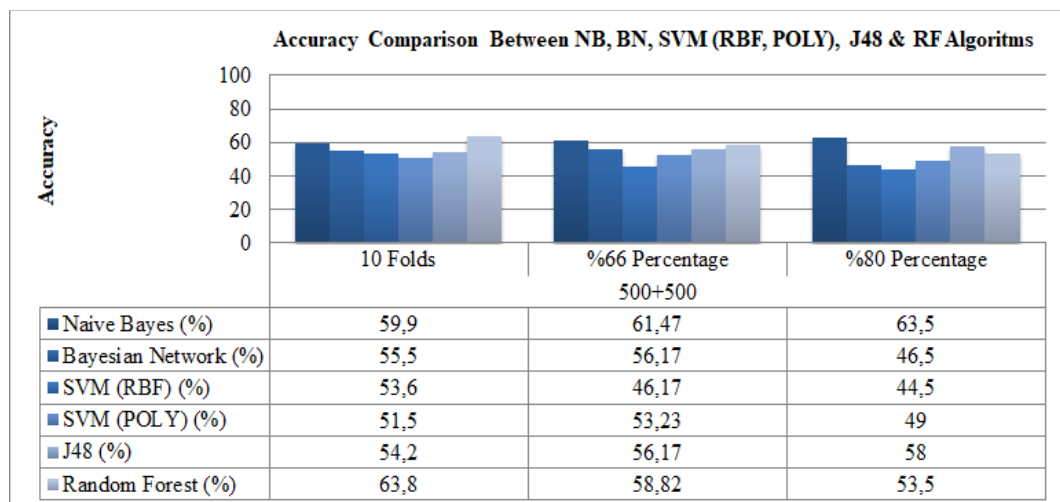


Figure 4-Accuracy comparison to 500 ham and 500 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms

As seen on Figure 5 for 1k+1k dataset, for 10 Folds method, best result is obtained by using RF algorithm. For 66% method, the best score is obtained by Random

Forest and for 80% method; the best score is obtained by Random Forest with 1000 ham + 1000 spam data.

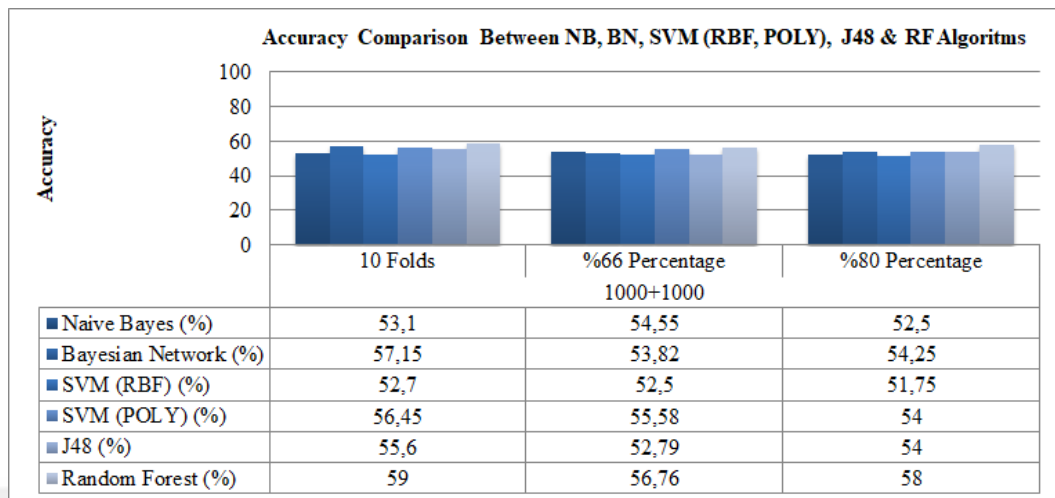


Figure 5-Accuracy comparison to 1000 ham and 1000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms

As seen on Figure 6, for 10 Folds method, best result is obtained by using Random Forest algorithm. For 66% method, best score is obtained by Naive Bayes and for 80% method; best score is obtained by Random Forest with 2000 ham + 2000 spam data.

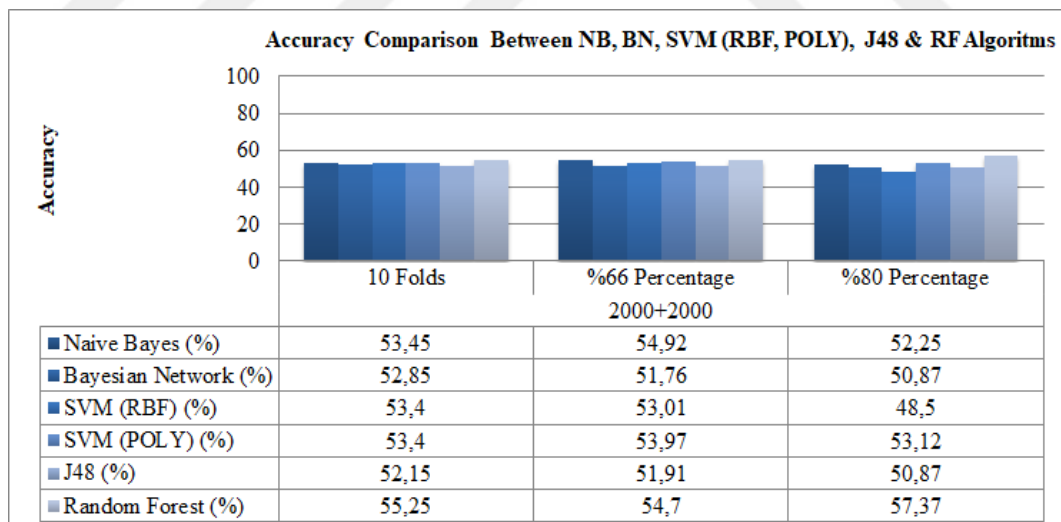


Figure 6-Accuracy comparison to 2000 ham and 2000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms

As seen on Figure 7, for 10 Folds method, best result is obtained by using Random Forest algorithm. For 66% method, best score is obtained by Random Forest and for

80% method; best score is obtained by Random Forest with 5000 ham + 5000 spam data.

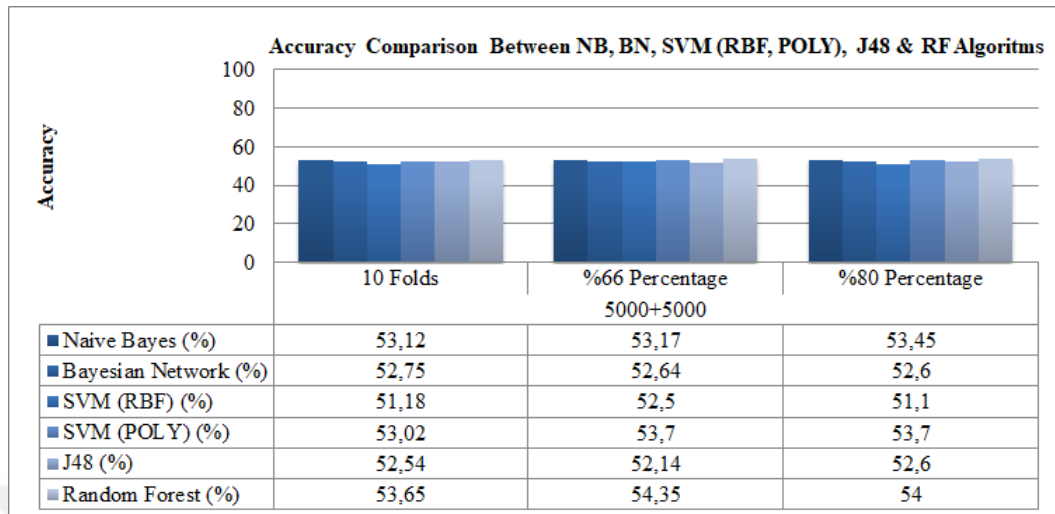


Figure 7-Accuracy comparison to 5000 ham and 5000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms

As seen on Figure 8, for 10 Folds method, best result is obtained by using Random Forest algorithm. For 66% method, best score is obtained by Random Forest and for 80% method, best score is obtained by Random Forest with 10000 ham + 10000 spam data.

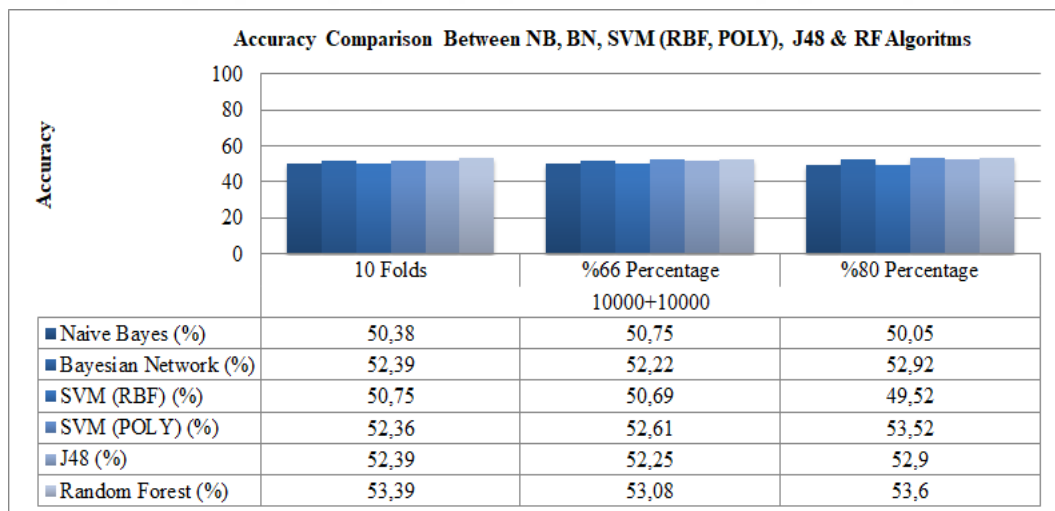


Figure 8-Accuracy comparison to 10000 ham and 10000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms

We also measured F-measure score for all tests (10-fold, 66%, and 80%) for all datasets (300+300, 500+500, 1k+1k, 2k+2k, 5k+5k, 10k+10k) and for all algorithms (NB, BN, SVM(RBF), SVM(POLY), J48, and RF). The detailed results are listed in Tables 7-15.

Table 7-The Results for Weighting F-Measures to 300&500 data sets between NB & BN

Data Sets	Test Type	Naive Bayes				Bayesian Network			
300+300	Cross Validation (10 Folds)	SPAM	HAM		0.620	SPAM	HAM		0.518
		240	60	SPAM		73	227	SPAM	
		168	132	HAM		62	238	HAM	
	Percentage (%66)	SPAM	HAM		0.618	SPAM	HAM		0.515
		82	18	SPAM		98	2	SPAM	
		60	44	HAM		97	7	HAM	
	Percentage (%80)	SPAM	HAM		0.650	SPAM	HAM		0.483
		46	10	SPAM		55	1	SPAM	
		32	32	HAM		61	3	HAM	
500+500	Cross Validation (10 Folds)	SPAM	HAM		0.599	SPAM	HAM		0.555
		452	48	SPAM		495	5	SPAM	
		353	147	HAM		440	60	HAM	
	Percentage (%66)	SPAM	HAM		0.615	SPAM	HAM		0.562
		171	12	SPAM		182	1	SPAM	
		119	38	HAM		148	9	HAM	
	Percentage (%80)	SPAM	HAM		0.635	SPAM	HAM		0.465
		103	8	SPAM		5	106	SPAM	
		65	24	HAM		1	68	HAM	

Table 8-The Results for Weighting F-Measures to 300&500 data sets between SVM (RBF & POLY)

Data Sets	Test Type	SVM (RBF)				SVM (POLY)			
300+300	Cross Validation (10 Folds)	SPAM	HAM		0.580	SPAM	HAM		0.617
		173	127	SPAM		125	175	SPAM	
		125	175	HAM		55	245	HAM	
	Percentage (%66)	SPAM	HAM		0.554	SPAM	HAM		0.657
		85	15	SPAM		46	54	SPAM	
		76	28	HAM		16	88	HAM	
	Percentage (%80)	SPAM	HAM		0.467	SPAM	HAM		0.675
		56	0	SPAM		25	31	SPAM	
		64	0	HAM		8	56	HAM	
500+500	Cross Validation (10 Folds)	SPAM	HAM		0.536	SPAM	HAM		0.515
		439	61	SPAM		359	141	SPAM	
		403	97	HAM		344	156	HAM	
	Percentage (%66)	SPAM	HAM		0.462	SPAM	HAM		0.532
		0	183	SPAM		53	130	SPAM	
		0	157	HAM		29	128	HAM	
	Percentage (%80)	SPAM	HAM		0.445	SPAM	HAM		0.490
		0	111	SPAM		26	85	SPAM	
		0	89	HAM		17	72	HAM	

Table 9-The Results for Weighting F-Measures to 300&500 data sets between J48 & RF

Data Sets	Test Type	J48				Random Forest			
300+300	Cross Validation (10 Folds)	SPAM	HAM		0.547	SPAM	HAM		0.680
		35	265	SPAM		157	143	SPAM	
		7	293	HAM		49	251	HAM	
	Percentage (%66)	SPAM	HAM		0.593	SPAM	HAM		0.711
		18	82	SPAM		54	46	SPAM	
		1	103	HAM		13	91	HAM	
	Percentage (%80)	SPAM	HAM		0.608	SPAM	HAM		0.733
		9	47	SPAM		30	26	SPAM	
		0	64	HAM		6	58	HAM	
500+500	Cross Validation (10 Folds)	SPAM	HAM		0.542	SPAM	HAM		0.638
		488	12	SPAM		26	294	SPAM	
		446	54	HAM		68	432	HAM	
	Percentage (%66)	SPAM	HAM		0.562	SPAM	HAM		0.588
		182	1	SPAM		74	109	SPAM	
		148	9	HAM		31	126	HAM	
	Percentage (%80)	SPAM	HAM		0.580	SPAM	HAM		0.535
		107	4	SPAM		39	72	SPAM	
		80	9	HAM		21	68	HAM	

Table 10-The Results for Weighting F-Measures to 1000&2000 data sets between NB & BN

Data Sets	Test Type	Naive Bayes			Bayesian Network				
1000+1000	Cross Validation (10 Folds)	SPAM	HAM		0.531	SPAM	HAM		0.572
		79	921	SPAM		988	12	SPAM	
		17	983	HAM		845	155	HAM	
	Percentage (%66)	SPAM	HAM		0.546	SPAM	HAM		0.538
		40	301	SPAM		335	6	SPAM	
		8	331	HAM		308	31	HAM	
	Percentage (%80)	SPAM	HAM		0.525	SPAM	HAM		0.543
		15	185	SPAM		200	0	SPAM	
		5	195	HAM		183	17	HAM	
2000+2000	Cross Validation (10 Folds)	SPAM	HAM		0.535	SPAM	HAM		0.529
		998	1002	SPAM		1994	6	SPAM	
		860	1140	HAM		1880	120	HAM	
	Percentage (%66)	SPAM	HAM		0.549	SPAM	HAM		0.518
		655	28	SPAM		683	0	SPAM	
		585	92	HAM		656	21	HAM	
	Percentage (%80)	SPAM	HAM		0.523	SPAM	HAM		0.509
		346	41	SPAM		387	0	SPAM	
		341	72	HAM		393	20	HAM	

Table 11-The Results for Weighting F-Measures to 1000&2000 data sets between SVM (RBF & POLY)

Data Sets	Test Type	SVM (RBF)			SVM (POLY)				
1000+1000	Cross Validation (10 Folds)	SPAM	HAM		0.527	SPAM	HAM	0.565	
		976	24	SPAM		884	116		SPAM
		922	78	HAM		755	245		HAM
	Percentage (%66)	SPAM	HAM		0.525	SPAM	HAM	0.556	
		331	10	SPAM		295	46		SPAM
		313	26	HAM		256	83		HAM
	Percentage (%80)	SPAM	HAM		0.518	SPAM	HAM	0.540	
		197	3	SPAM		169	31		SPAM
		190	10	HAM		153	47		HAM
2000+2000	Cross Validation (10 Folds)	SPAM	HAM		0.534	SPAM	HAM	0.534	
		1568	432	SPAM		1824	176		SPAM
		1432	568	HAM		1688	312		HAM
	Percentage (%66)	SPAM	HAM		0.530	SPAM	HAM	0.540	
		667	16	SPAM		634	49		SPAM
		623	54	HAM		577	100		HAM
	Percentage (%80)	SPAM	HAM		0.485	SPAM	HAM	0.531	
		387	0	SPAM		365	22		SPAM
		412	1	HAM		353	60		HAM

Table 12-The Results for Weighting F-Measures to 1000&2000 data sets between J48 & RF

Data Sets	Test Type	J48				Random Forest			
1000+1000	Cross Validation (10 Folds)	SPAM	HAM		0.556	SPAM	HAM		0.590
		986	14	SPAM		278	732	SPAM	
		874	126	HAM		98	902	HAM	
	Percentage (%66)	SPAM	HAM		0.528	SPAM	HAM		0.568
		334	7	SPAM		91	250	SPAM	
		341	25	HAM		44	295	HAM	
	Percentage (%80)	SPAM	HAM		0.540	SPAM	HAM		0.580
		196	4	SPAM		55	145	SPAM	
		180	20	HAM		23	117	HAM	
2000+2000	Cross Validation (10 Folds)	SPAM	HAM		0.522	SPAM	HAM		0.553
		1985	15	SPAM		355	1645	SPAM	
		1899	101	HAM		145	1855	HAM	
	Percentage (%66)	SPAM	HAM		0.519	SPAM	HAM		0.547
		677	6	SPAM		104	579	SPAM	
		648	29	HAM		37	640	HAM	
	Percentage (%80)	SPAM	HAM		0.509	SPAM	HAM		0.574
		385	2	SPAM		68	319	SPAM	
		391	22	HAM		22	391	HAM	

Table 13-The Results for Weighting F-Measures to 5000&1000 data sets between NB & BN

Data Sets	Test Type	Naive Bayes			Bayesian Network				
5000+5000	Cross Validation (10 Folds)	SPAM	HAM		0.531	SPAM	HAM	0.528	
		4968	32	SPAM		4990	10		SPAM
		4656	344	HAM		4715	285		HAM
	Percentage (%66)	SPAM	HAM		0.318	SPAM	HAM	0.526	
		1679	15	SPAM		1691	3		SPAM
		1577	129	HAM		1607	99		HAM
	Percentage (%80)	SPAM	HAM		0.535	SPAM	HAM	0.526	
		990	12	SPAM		999	3		SPAM
		919	79	HAM		945	53		HAM
10000+10000	Cross Validation (10 Folds)	SPAM	HAM		0.504	SPAM	HAM	0.524	
		121	9879	SPAM		9985	15		SPAM
		44	9956	HAM		9507	493		HAM
	Percentage (%66)	SPAM	HAM		0.508	SPAM	HAM	0.522	
		63	3331	SPAM		3391	3		SPAM
		18	3388	HAM		3246	160		HAM
	Percentage (%80)	SPAM	HAM		0.501	SPAM	HAM	0.529	
		29	1990	SPAM		2016	3		SPAM
		8	1973	HAM		1880	101		HAM

Table 14-The Results for Weighting F-Measures to 5000&10000 data sets between SVM (RBF & POLY)

Data Sets	Test Type	SVM (RBF)				SVM (POLY)			
5000+5000	Cross Validation (10 Folds)	SPAM	HAM		0.512	SPAM	HAM		0.530
		579	4421	SPAM		4852	148	SPAM	
		461	4539	HAM		4550	450	HAM	
	Percentage (%66)	SPAM	HAM		0.525	SPAM	HAM		0.537
		1668	26	SPAM		1657	37	SPAM	
		1589	117	HAM		15	169	HAM	
	Percentage (%80)	SPAM	HAM		0.511	SPAM	HAM		0.537
		30	972	SPAM		976	26	SPAM	
		6	992	HAM		900	98	HAM	
10000+10000	Cross Validation (10 Folds)	SPAM	HAM		0.508	SPAM	HAM		0.524
		7974	2026	SPAM		9839	161	SPAM	
		7824	2176	HAM		9366	634	HAM	
	Percentage (%66)	SPAM	HAM		0.507	SPAM	HAM		0.526
		3384	10	SPAM		3348	46	SPAM	
		3343	63	HAM		3176	230	HAM	
	Percentage (%80)	SPAM	HAM		0.495	SPAM	HAM		0.535
0		2019	SPAM	1978		41	SPAM		
0		1981	HAM	1818		163	HAM		

Table 15-The Results for Weighting F-Measures to 5000&10000 data sets between J48 & RF

Data Sets	Test Type	J48				Random Forest			
5000+5000	Cross Validation (10 Folds)	SPAM	HAM		0.525	SPAM	HAM		0.537
		4994	6	SPAM		4804	196	SPAM	
		4740	260	HAM		4439	561	HAM	
	Percentage (%66)	SPAM	HAM		0.521	SPAM	HAM		0.544
		1692	2	SPAM		1640	54	SPAM	
		1625	81	HAM		1498	208	HAM	
	Percentage (%80)	SPAM	HAM		0.526	SPAM	HAM		0.540
		999	3	SPAM		960	42	SPAM	
		945	53	HAM		878	120	HAM	
10000+10000	Cross Validation (10 Folds)	SPAM	HAM		0.524	SPAM	HAM		0.534
		9975	25	SPAM		9780	220	SPAM	
		9496	504	HAM		9102	898	HAM	
	Percentage (%66)	SPAM	HAM		0.523	SPAM	HAM		0.531
		3390	4	SPAM		3311	83	SPAM	
		3243	163	HAM		3107	299	HAM	
	Percentage (%80)	SPAM	HAM		0.529	SPAM	HAM		0.536
		2015	4	SPAM		1965	54	SPAM	
		1880	101	HAM		1802	179	HAM	

F-measure results are compared as bar graphs in the following Figures 9-14. Here are our observations:

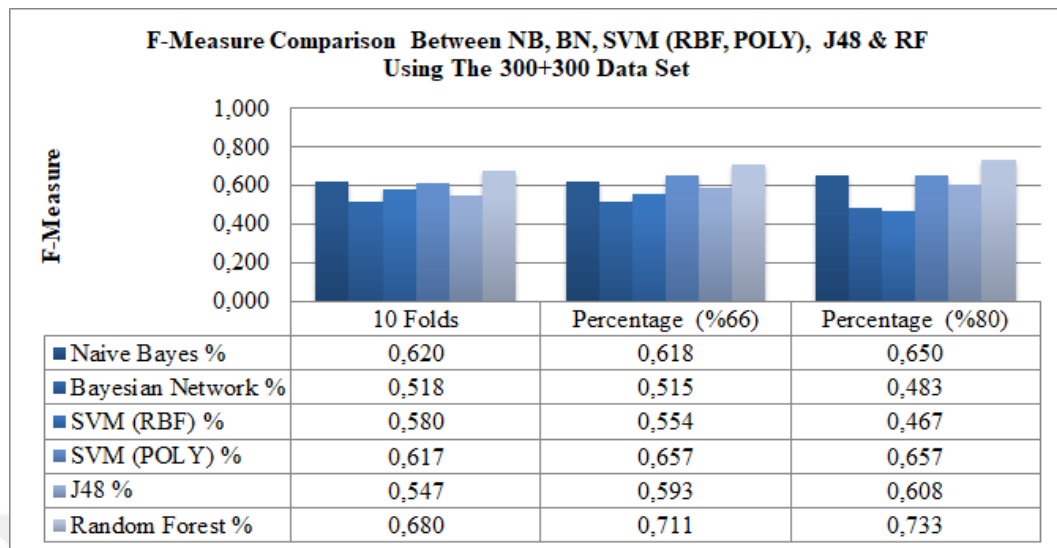


Figure 9-F-Measure comparison to 300 ham and 300 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms

As seen on Figure 9, the best score is the Percentage %80 result of the Random Forest algorithm with 300 ham + 300 spam data.

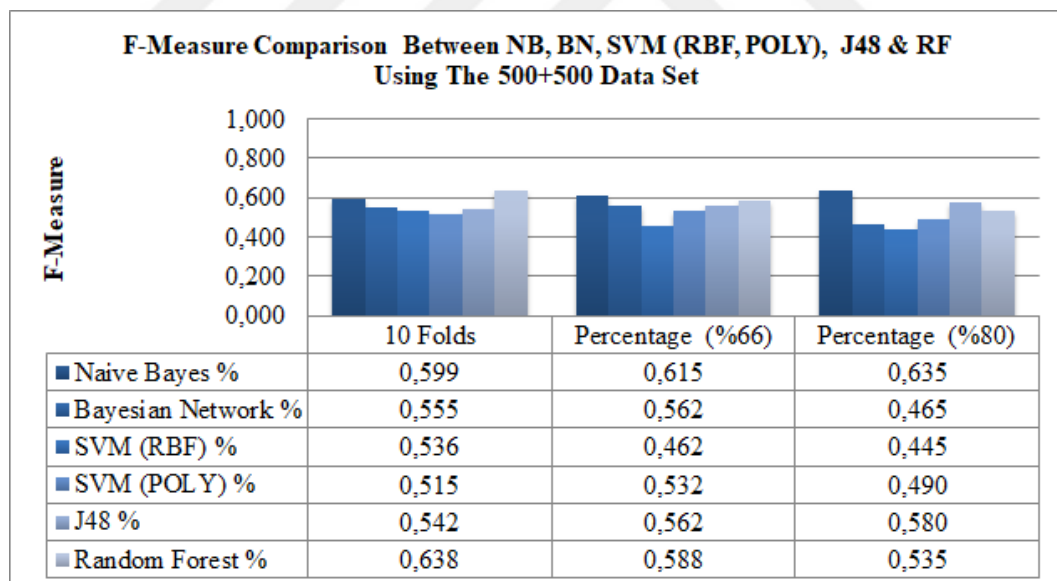


Figure 10-F-Measure comparison to 500 ham and 500 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms

As seen on Figure 10, the best score is the 10 Folds result of the Random Forest algorithm with 500 ham + 500 spam data.

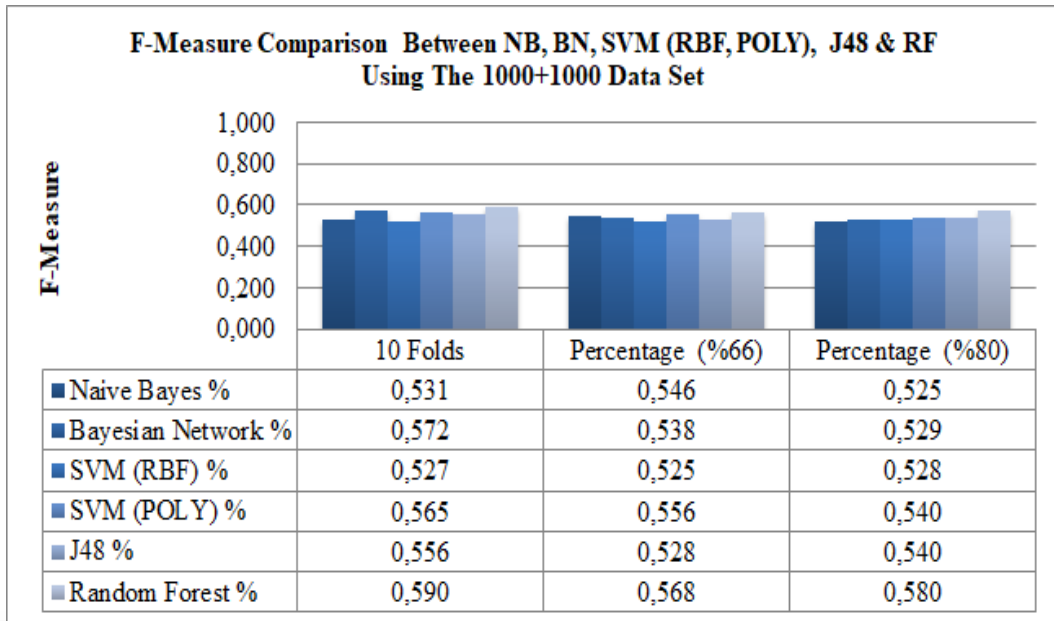


Figure 11-F-Measure comparison to 1000 ham and 1000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms

As seen on Figure 11, the best score is the 10 Folds result of the Random Forest algorithm with 1000 ham + 1000 spam data.

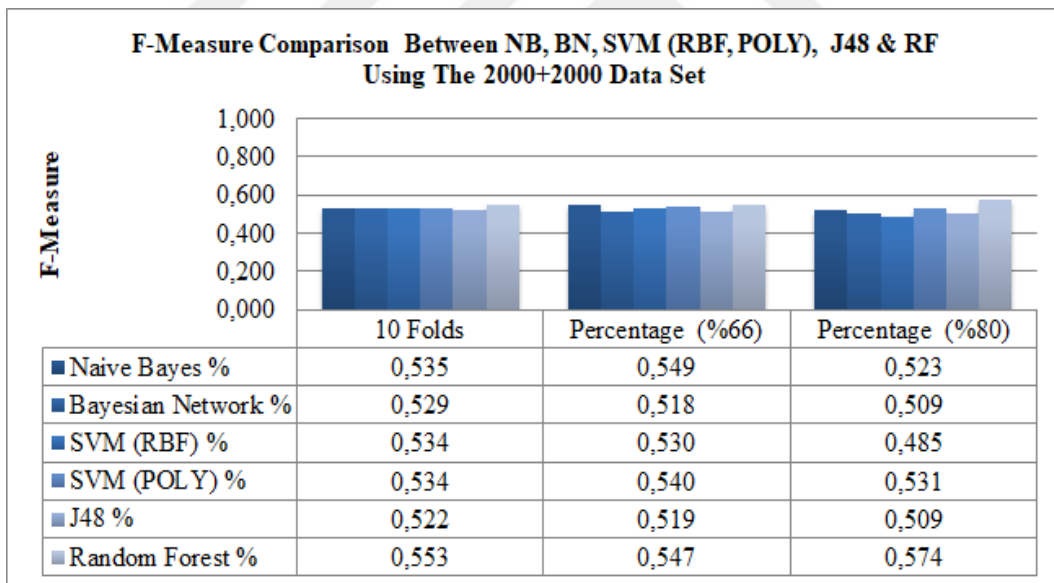


Figure 12-F-Measure comparison to 2000 ham and 2000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms

As seen on Figure 12, the best score is the Percentage %80 result of the Random Forest algorithm with 2000 ham + 2000 spam data.

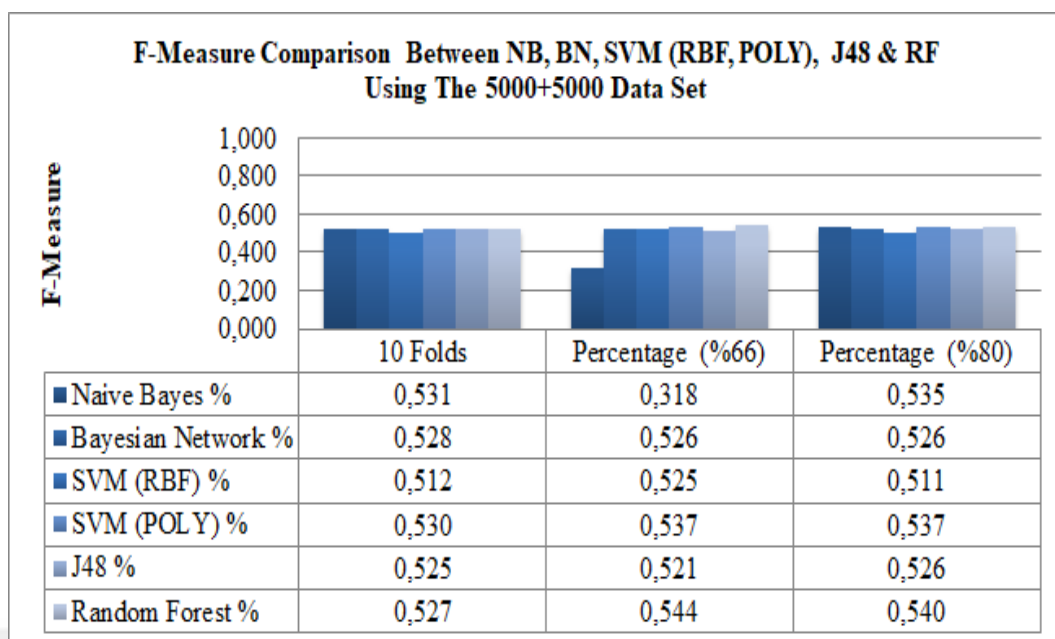


Figure 13-F-Measure comparison to 5000 ham and 5000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms

As seen on Figure 13, the best score is the Percentage %66 result of the Random Forest algorithm with 5000 ham + 5000 spam data.

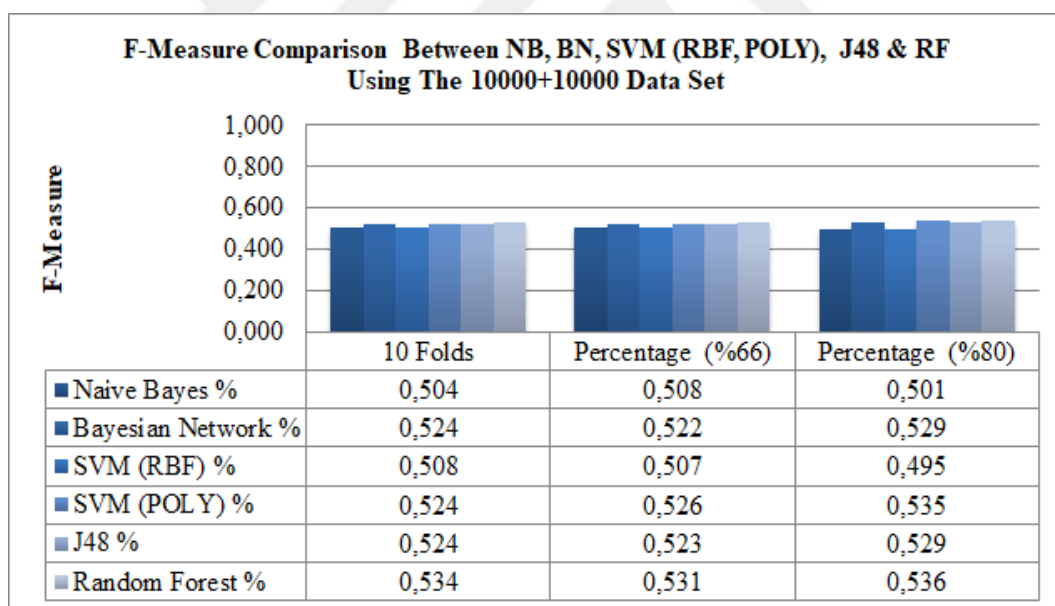


Figure 14-F-Measure comparison to 10000 ham and 10000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms

As seen on Figure 14, the best score is the Percentage %80 result of the Random Forest algorithm with 10000 ham + 10000 spam data.

4.3.2 SciKit Learn with TF-IDF

We also tested SciKit Learn library's TF-IDF based vector representation and found the following results (Table 16 and Figure 15).

Table 16-SciKit Learn Tools Results

Data Set Ham and Spam	SciKit Learn (%)
300	92.000
500	93.000
1000	93.250
2000	95.330
5000	96.310
10000	96.845

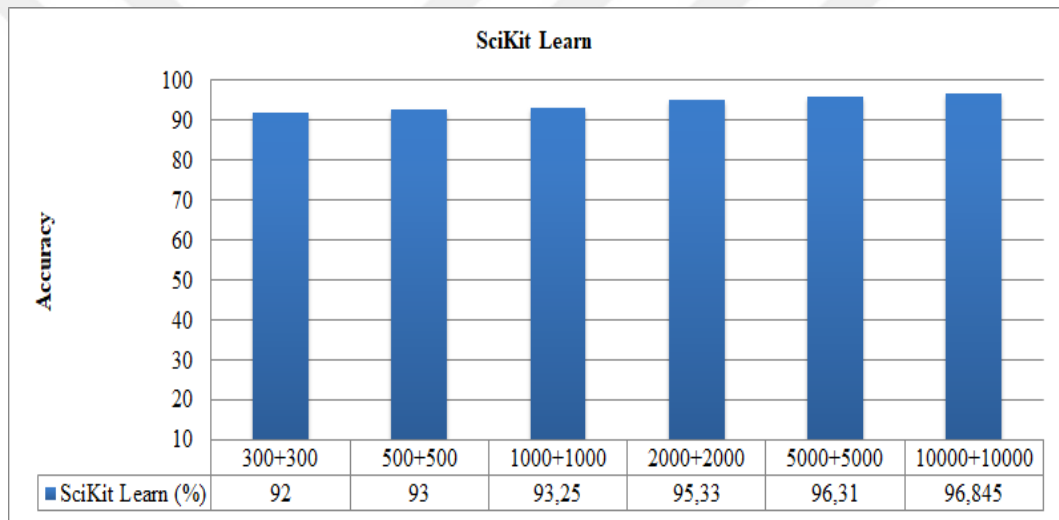


Figure 15-Different data sets and results for SciKit Learn Tools

As seen on Figure 15, SciKit Learn tools accuracy results for all datasets. As the size of the data set increased, the success rate displayed a steady increase in percentage.

4.3.3 Word2Vec

Here we tested Word2Vec-based vector representation on email datasets. The dataset is divided in the same way as shown the table above. Algorithm results are also compared with three different methods, 10 folds (average of 10 train / test data set), 66% and 80%, where we obtained different results. The best score for each data set is highlighted in bold. We depict these results below:

Table 17-Test Results for Word2Vec Accuracy

Data Sets	Process	Naive Bayes (%)	Bayesian Network (%)	SVM (RBF) (%)	SVM (POLY) (%)	J48 (%)	Random Forest (%)
300 + 300	10 Folds	75.16	85.16	90.5	95	78.83	92.66
	%66 Percentage	78.92	86.76	90.19	96.56	83.82	92.64
	%80 Percentage	71.66	80.83	88.33	98.33	82.5	94.16
500 + 500	10 Folds	69.4	83.8	89.3	93.2	79.1	92.9
	%66 Percentage	68.52	83.82	86.17	91.47	79.7	91.76
	%80 Percentage	69	82	86.5	93.5	78	93.5
1000 + 1000	10 Folds	72.5	84.45	90.8	95.75	81	93.85
	%66 Percentage	70.14	83.82	91.02	96.47	80.44	92.35
	%80 Percentage	69.75	85.5	90.75	96.25	80.5	93.25
2000 + 2000	10 Folds	75.2	85.17	91.87	95.65	83.2	94.4
	%66 Percentage	77.5	84.77	91.98	95.73	84.48	94.92
	%80 Percentage	76.75	85.5	92.5	96.25	83.25	94.87
5000 + 5000	10 Folds	74.5	85.65	92.9	96.02	86.78	95.55
	%66 Percentage	74.82	85.67	91.97	95.64	84.85	95
	%80 Percentage	73.75	84.8	92.05	95.85	85.05	95.05
10000 + 10000	10 Folds	75.77	86.3	94.03	96.26	88.03	96.19
	%66 Percentage	75.63	86.11	93.58	95.88	87.05	95.45
	%80 Percentage	76.37	86.35	93.95	96.32	87.2	95.77

As seen on Figure 16, for 10 Folds method, best result is obtained by using Random Forest algorithm. For 66% method, the best score is obtained by SVM (POLY) and for 80% method, the best score is obtained by SVM (Poly) with 300 ham + 300 spam data.

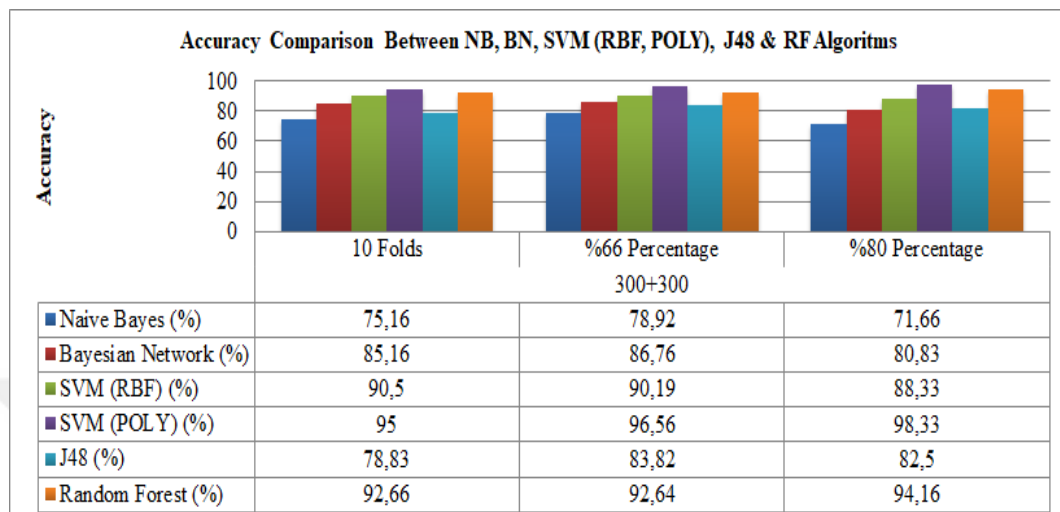


Figure 16-Accuracy comparison to 300 ham and 300 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms

As seen on Figure 17, for 10 Folds method, best result is obtained by using SVM (Poly) algorithm. For 66% method, the best score is obtained by Random Forest and for 80% method, the best score is same with SVM (Poly) and Random Forest with 500 ham + 500 spam data.

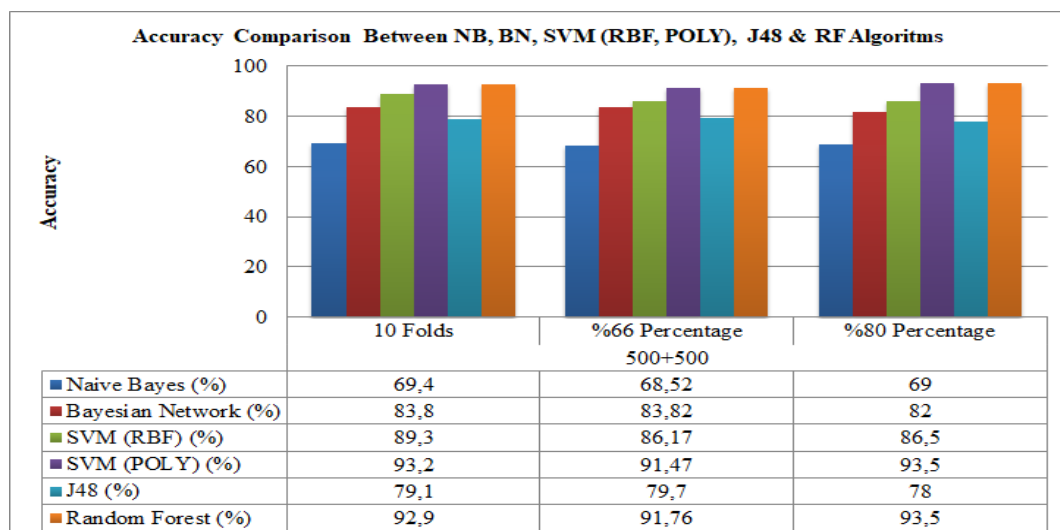


Figure 17-Accuracy comparison to 500 ham and 500 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms

As seen on Figure 18, for 10 Folds method, the best result is obtained by using SVM (Poly) algorithm. For 66% method, the best score is obtained by SVM (Poly) and for 80% method; the best score is obtained by SVM (Poly) with 1000 ham + 1000 spam data.

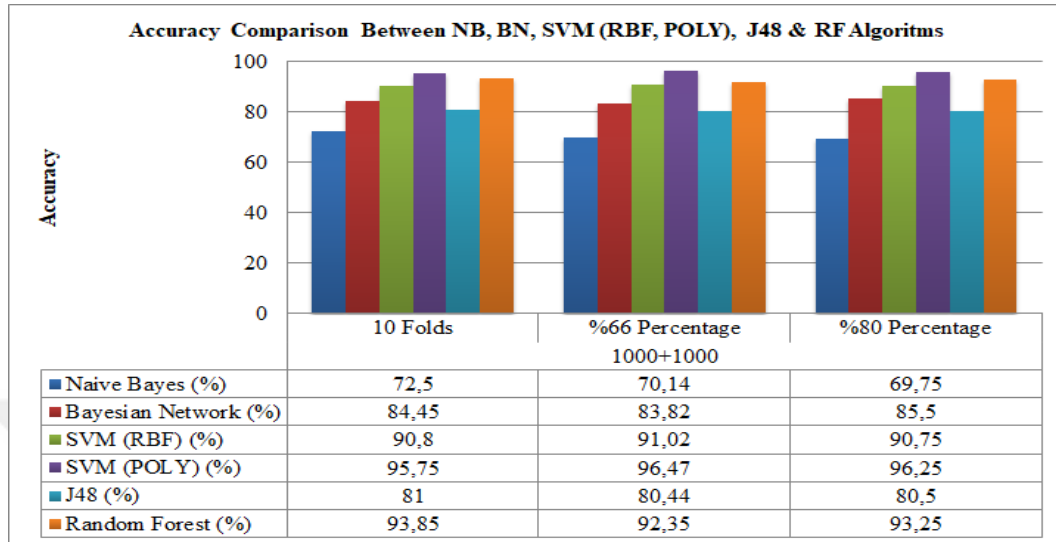


Figure 18-Accuracy comparison to 1000 ham and 1000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms

As seen on Figure 19, for 10 Folds method, best result is obtained by using SVM (Poly) algorithm. For 66% method, the best score is obtained by SVM (Poly) and for 80% method; the best score is obtained by SVM (Poly) with 2000 ham + 2000 spam data.

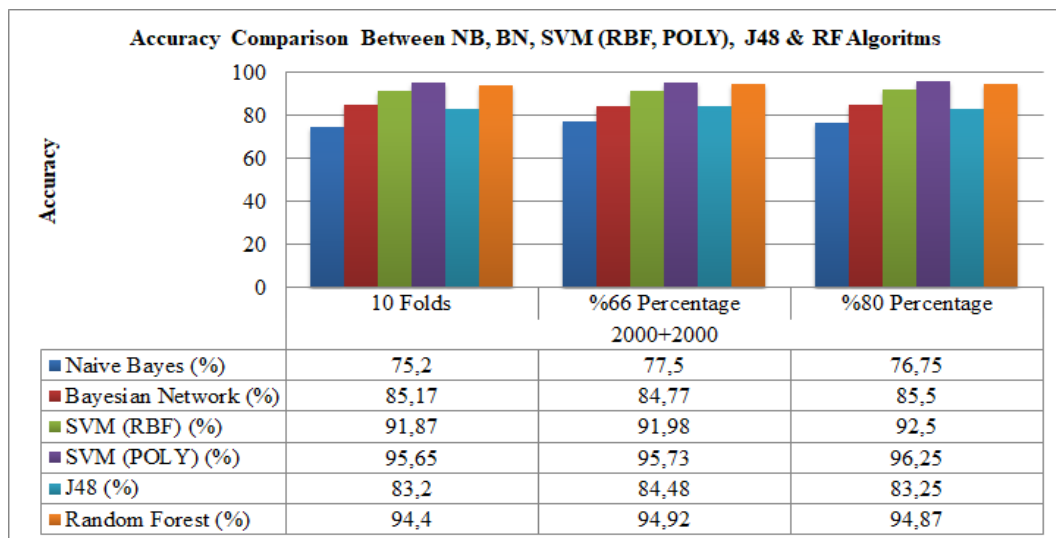


Figure 19-Accuracy comparison to 2000 ham and 2000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms

As seen on Figure 20, for 10 Folds method, the best result is obtained by using SVM (Poly) algorithm. For 66% method, the best score is obtained by SVM (Poly) and for 80% method; the best score is obtained by SVM (Poly) with 5000 ham + 5000 spam data.

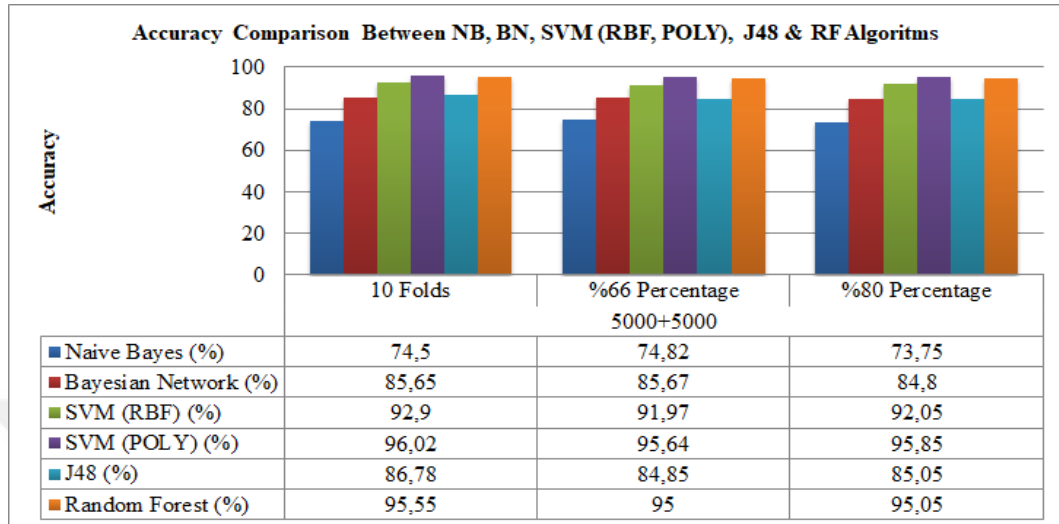


Figure 20-Accuracy comparison to 5000 ham and 5000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms

As seen on Figure 21, for 10 Folds method, the best result is obtained by using SVM (Poly) algorithm. For 66% method, the best score is obtained by SVM (Poly) and for 80% method, the best score is obtained by SVM (Poly) with 10000 ham + 10000 spam data.

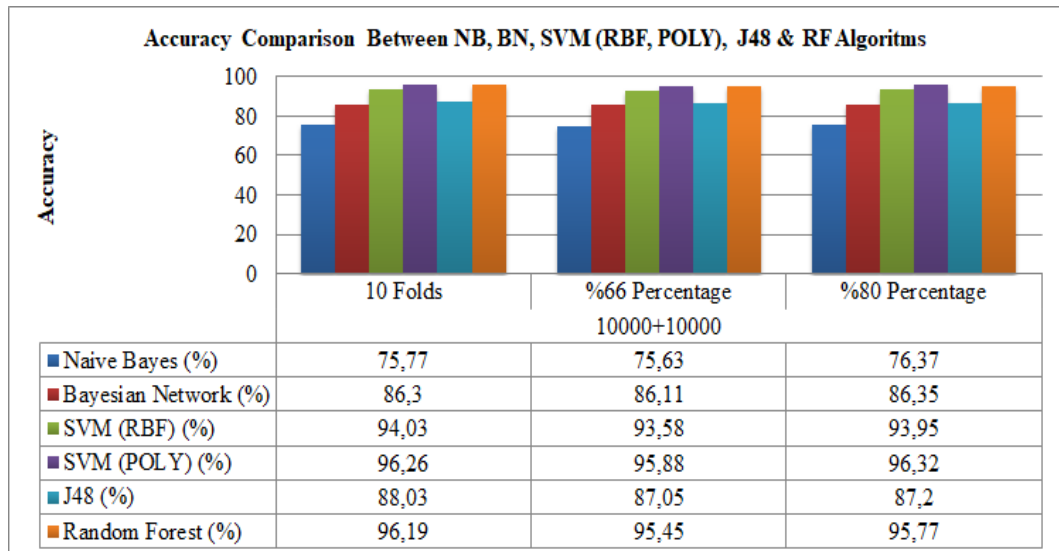


Figure 21-Accuracy comparison to 10000 ham and 10000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms

Table 18-26 lists F-measure results for Word2Vec representation. The best scores are highlighted.

Table 18-The Results for Word2Vec F-Measures to 300&500 data sets between NB & BN

Data Sets	Test Type	Naive Bayes				Bayesian Network			
300+300	Cross Validation (10 Folds)	SPAM	HAM		0.752	SPAM	HAM		0.852
		187	113	SPAM		254	46	SPAM	
		36	264	HAM		43	257	HAM	
	Percentage (%66)	SPAM	HAM		0.789	SPAM	HAM		0.868
		65	35	SPAM		85	15	SPAM	
		8	96	HAM		12	92	HAM	
	Percentage (%80)	SPAM	HAM		0.717	SPAM	HAM		0.808
		31	25	SPAM		43	13	SPAM	
		9	55	HAM		10	54	HAM	
500+500	Cross Validation (10 Folds)	SPAM	HAM		0.694	SPAM	HAM		0.838
		276	224	SPAM		401	99	SPAM	
		82	418	HAM		63	437	HAM	
	Percentage (%66)	SPAM	HAM		0.682	SPAM	HAM		0.838
		95	88	SPAM		154	29	SPAM	
		19	138	HAM		26	131	HAM	
	Percentage (%80)	SPAM	HAM		0.690	SPAM	HAM		0.820
		58	53	SPAM		87	24	SPAM	
		9	80	HAM		12	77	HAM	

Table 19-The Results for Word2Vec F-Measures to 300&500 data sets between SVM (RBF&POLY)

Data Sets	Test Type	SVM (RBF)			SVM (POLY)				
300+300	Cross Validation (10 Folds)	SPAM	HAM		0.905	SPAM	HAM	0.950	
		255	41	SPAM		286	14		SPAM
		16	284	HAM		16	284		HAM
	Percentage (%66)	SPAM	HAM		0.902	SPAM	HAM	0.966	
		89	11	SPAM		98	2		SPAM
		9	95	HAM		5	99		HAM
	Percentage (%80)	SPAM	HAM		0.883	SPAM	HAM	0.983	
		47	9	SPAM		56	0		SPAM
		5	59	HAM		2	62		HAM
500+500	Cross Validation (10 Folds)	SPAM	HAM		0.893	SPAM	HAM	0.932	
		426	74	SPAM		469	31		SPAM
		33	467	HAM		37	463		HAM
	Percentage (%66)	SPAM	HAM		0.862	SPAM	HAM	0.915	
		147	36	SPAM		168	15		SPAM
		11	146	HAM		14	143		HAM
	Percentage (%80)	SPAM	HAM		0.865	SPAM	HAM	0.935	
		91	20	SPAM		104	7		SPAM
		7	82	HAM		6	83		HAM

Table 20-The Results for Word2Vec F-Measures to 300&500 data sets between J48 & RF

Data Sets	Test Type	J48				Random Forest			
300+300	Cross Validation (10 Folds)	SPAM	HAM		0.788	SPAM	HAM		0.927
		240	60	SPAM		285	15	SPAM	
		67	233	HAM		29	271	HAM	
	Percentage (%66)	SPAM	HAM		0.838	SPAM	HAM		0.926
		82	18	SPAM		96	4	SPAM	
		15	89	HAM		11	93	HAM	
	Percentage (%80)	SPAM	HAM		0.825	SPAM	HAM		0.942
		45	11	SPAM		55	1	SPAM	
		10	54	HAM		6	58	HAM	
500+500	Cross Validation (10 Folds)	SPAM	HAM		0.791	SPAM	HAM		0.929
		407	93	SPAM		476	24	SPAM	
		116	384	HAM		47	453	HAM	
	Percentage (%66)	SPAM	HAM		0.797	SPAM	HAM		0.918
		151	32	SPAM		175	8	SPAM	
		37	120	HAM		20	137	HAM	
	Percentage (%80)	SPAM	HAM		0.780	SPAM	HAM		0.935
		88	23	SPAM		106	5	SPAM	
		21	68	HAM		8	81	HAM	

Table 21-The Results for Word2Vec F-Measures to 1000&2000 data sets between NB & BN

Data Sets	Test Type	Naive Bayes			Bayesian Network				
1000+1000	Cross Validation (10 Folds)	SPAM	HAM		0.725	SPAM	HAM		0.845
		593	407	SPAM		809	191	SPAM	
		143	857	HAM		120	880	HAM	
	Percentage (%66)	SPAM	HAM		0.701	SPAM	HAM		0.838
		204	137	SPAM		279	62	SPAM	
		66	273	HAM		48	291	HAM	
	Percentage (%80)	SPAM	HAM		0.698	SPAM	HAM		0.855
		115	85	SPAM		166	34	SPAM	
		36	164	HAM		24	176	HAM	
2000+2000	Cross Validation (10 Folds)	SPAM	HAM		0.752	SPAM	HAM		0.852
		1268	732	SPAM		1645	355	SPAM	
		260	1760	HAM		238	1762	HAM	
	Percentage (%66)	SPAM	HAM		0.775	SPAM	HAM		0.848
		435	248	SPAM		546	137	SPAM	
		58	619	HAM		70	607	HAM	
	Percentage (%80)	SPAM	HAM		0.768	SPAM	HAM		0.855
		241	146	SPAM		310	77	SPAM	
		40	373	HAM		39	374	HAM	

Table 22-The Results for Word2Vec F-Measures to 1000&2000 data sets between SVM (RBF&POLY)

Data Sets	Test Type	SVM (RBF)			SVM (POLY)				
1000+1000	Cross Validation (10 Folds)	SPAM	HAM		0.908	SPAM	HAM		0.958
		873	127	SPAM		961	39	SPAM	
		57	943	HAM		46	954	HAM	
	Percentage (%66)	SPAM	HAM		0.910	SPAM	HAM		0.965
		298	43	SPAM		329	12	SPAM	
		18	321	HAM		12	327	HAM	
	Percentage (%80)	SPAM	HAM		0.908	SPAM	HAM		0.963
		174	26	SPAM		192	8	SPAM	
		11	189	HAM		7	193	HAM	
2000+2000	Cross Validation (10 Folds)	SPAM	HAM		0.919	SPAM	HAM		0.957
		1805	195	SPAM		1924	76	SPAM	
		130	1870	HAM		98	1902	HAM	
	Percentage (%66)	SPAM	HAM		0.920	SPAM	HAM		0.957
		618	65	SPAM		661	22	SPAM	
		44	633	HAM		36	641	HAM	
	Percentage (%80)	SPAM	HAM		0.925	SPAM	HAM		0.963
		349	38	SPAM		376	11	SPAM	
		22	391	HAM		19	394	HAM	

Table 23-The Results for Word2Vec F-Measures to 1000&2000 data sets between J48 & RF

Data Sets	Test Type	J48			Random Forest				
1000+1000	Cross Validation (10 Folds)	SPAM	HAM		0.810	SPAM	HAM		0.939
		826	174	SPAM		952	48	SPAM	
		206	794	HAM		75	925	HAM	
	Percentage (%66)	SPAM	HAM		0.804	SPAM	HAM		0.924
		274	67	SPAM		323	18	SPAM	
		66	273	HAM		34	305	HAM	
	Percentage (%80)	SPAM	HAM		0.805	SPAM	HAM		0.933
		167	33	SPAM		191	8	SPAM	
		45	155	HAM		18	182	HAM	
2000+2000	Cross Validation (10 Folds)	SPAM	HAM		0.832	SPAM	HAM		0.944
		1677	323	SPAM		1917	83	SPAM	
		349	1651	HAM		141	1859	HAM	
	Percentage (%66)	SPAM	HAM		0.845	SPAM	HAM		0.929
		578	105	SPAM		660	23	SPAM	
		106	571	HAM		46	631	HAM	
	Percentage (%80)	SPAM	HAM		0.833	SPAM	HAM		0.948
		311	76	SPAM		371	16	SPAM	
		58	355	HAM		25	388	HAM	

Table 24-The Results for Word2Vec F-Measures to 5000&10000 data sets between NB & BN

Data Sets	Test Type	Naive Bayes				Bayesian Network			
5000+5000	Cross Validation (10 Folds)	SPAM	HAM		0.745	SPAM	HAM		0.857
		3099	1901	SPAM		4115	885	SPAM	
		649	4351	HAM		550	4450	HAM	
	Percentage (%66)	SPAM	HAM		0.748	SPAM	HAM		0.857
		1059	635	SPAM		1412	282	SPAM	
		221	1485	HAM		205	1501	HAM	
	Percentage (%80)	SPAM	HAM		0.738	SPAM	HAM		0.847
		619	383	SPAM		824	178	SPAM	
		142	856	HAM		126	872	HAM	
10000+10000	Cross Validation (10 Folds)	SPAM	HAM		0.758	SPAM	HAM		0.863
		6350	3650	SPAM		8347	1653	SPAM	
		1196	8804	HAM		1087	8913	HAM	
	Percentage (%66)	SPAM	HAM		0.756	SPAM	HAM		0.861
		5143	2136	SPAM		2846	548	SPAM	
		6800	399	HAM		396	3010	HAM	
	Percentage (%80)	SPAM	HAM		0.764	SPAM	HAM		0.864
		1304	715	SPAM		1701	318	SPAM	
		230	1751	HAM		228	1753	HAM	

Table 25-The Results for Word2Vec F-Measures to 5000&10000 data sets between SVM (RBF & POLY)

Data Sets	Test Type	SVM (RBF)				SVM (POLY)			
5000+5000	Cross Validation (10 Folds)	SPAM	HAM		0.929	SPAM	HAM		0.960
		4612	388	SPAM		4844	156	SPAM	
		322	4678	HAM		242	4758	HAM	
	Percentage (%66)	SPAM	HAM		0.927	SPAM	HAM		0.956
		1535	159	SPAM		1631	63	SPAM	
		114	1592	HAM		85	1621	HAM	
	Percentage (%80)	SPAM	HAM		0.921	SPAM	HAM		0.959
		909	93	SPAM		969	33	SPAM	
		66	932	HAM		50	948	HAM	
10000+10000	Cross Validation (10 Folds)	SPAM	HAM		0.940	SPAM	HAM		0.963
		9480	520	SPAM		9712	288	SPAM	
		673	9325	HAM		459	9541	HAM	
	Percentage (%66)	SPAM	HAM		0.936	SPAM	HAM		0.959
		3212	182	SPAM		3295	99	SPAM	
		254	3152	HAM		181	3225	HAM	
	Percentage (%80)	SPAM	HAM		0.940	SPAM	HAM		0.963
		1921	98	SPAM		1971	48	SPAM	
		144	1837	HAM		99	1882	HAM	

Table 26-The Results for Word2Vec F-Measures to 5000&10000 data sets between J48 & RF

Data Sets	Test Type	J48				Random Forest			
5000+5000	Cross Validation (10 Folds)	SPAM	HAM		0.868	SPAM	HAM		0.956
		4347	653	SPAM		4855	173	SPAM	
		669	4331	HAM		272	4728	HAM	
	Percentage (%66)	SPAM	HAM		0.849	SPAM	HAM		0.950
		1432	262	SPAM		1636	58	SPAM	
		253	1453	HAM		112	1594	HAM	
	Percentage (%80)	SPAM	HAM		0.851	SPAM	HAM		0.951
		864	138	SPAM		963	39	SPAM	
		161	857	HAM		60	938	HAM	
10000+10000	Cross Validation (10 Folds)	SPAM	HAM		0.880	SPAM	HAM		0.962
		8835	1165	SPAM		9681	319	SPAM	
		1228	8772	HAM		443	9557	HAM	
	Percentage (%66)	SPAM	HAM		0.871	SPAM	HAM		0.955
		2962	432	SPAM		3276	118	SPAM	
		448	2958	HAM		191	3215	HAM	
	Percentage (%80)	SPAM	HAM		0.872	SPAM	HAM		0.958
		1782	237	SPAM		1955	64	SPAM	
		275	1706	HAM		105	1876	HAM	

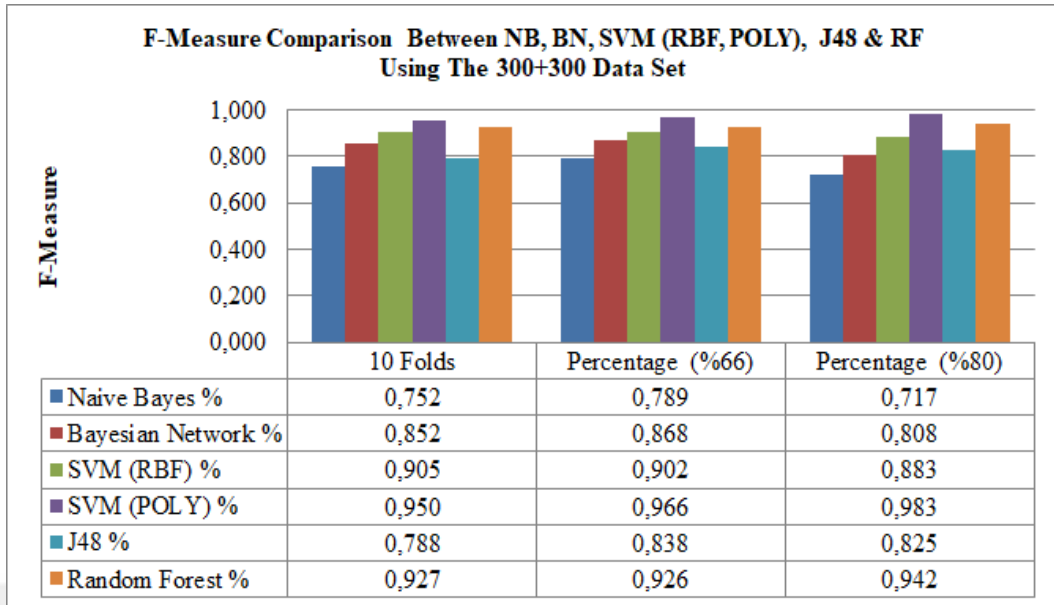


Figure 22-F-Measure comparison to 300 ham and 300 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms

As seen on Figure 22, the best score is the Percentage %80 result of the SVM (Poly) algorithm with 300 ham + 300 spam data.

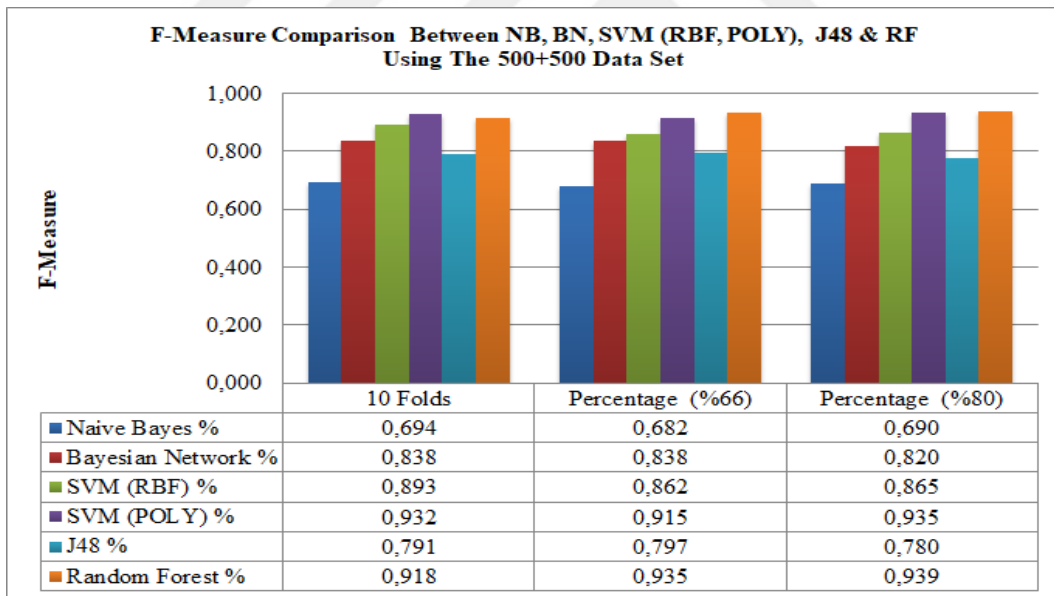


Figure 23-F-Measure comparison to 500 ham and 500 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms

As seen on Figure 23, the best score is the Percentage %80 result of the SVM (Poly) and Random Forest algorithm with 500 ham + 500 spam data.

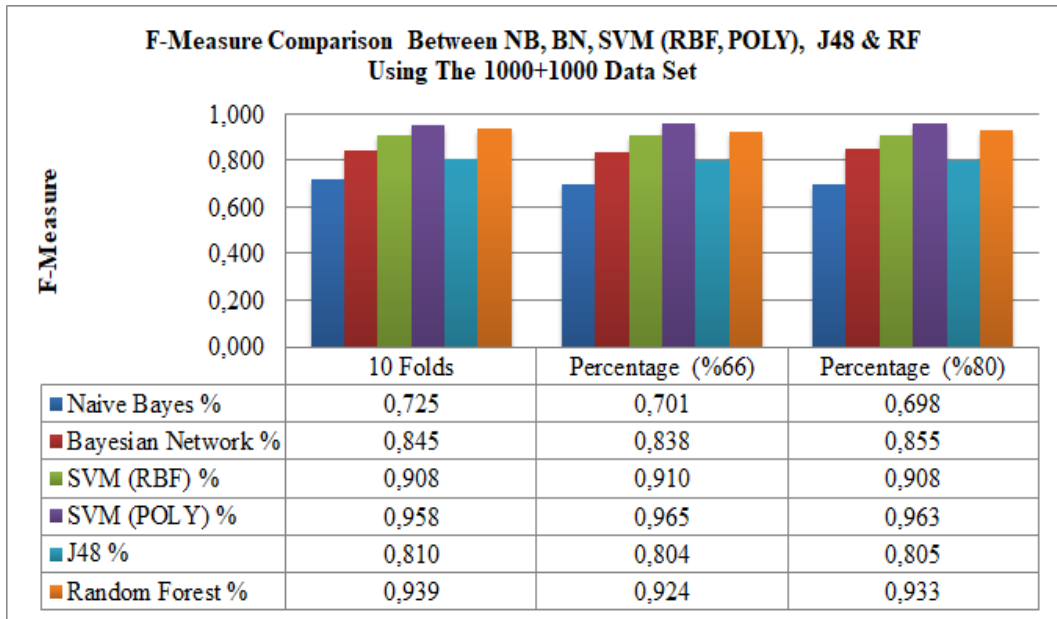


Figure 24-F-Measure comparison to 1000 ham and 1000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms

As seen on Figure 24, the best score is the Percentage %66 result of the SVM (Poly) algorithm with 1000 ham + 1000 spam data.

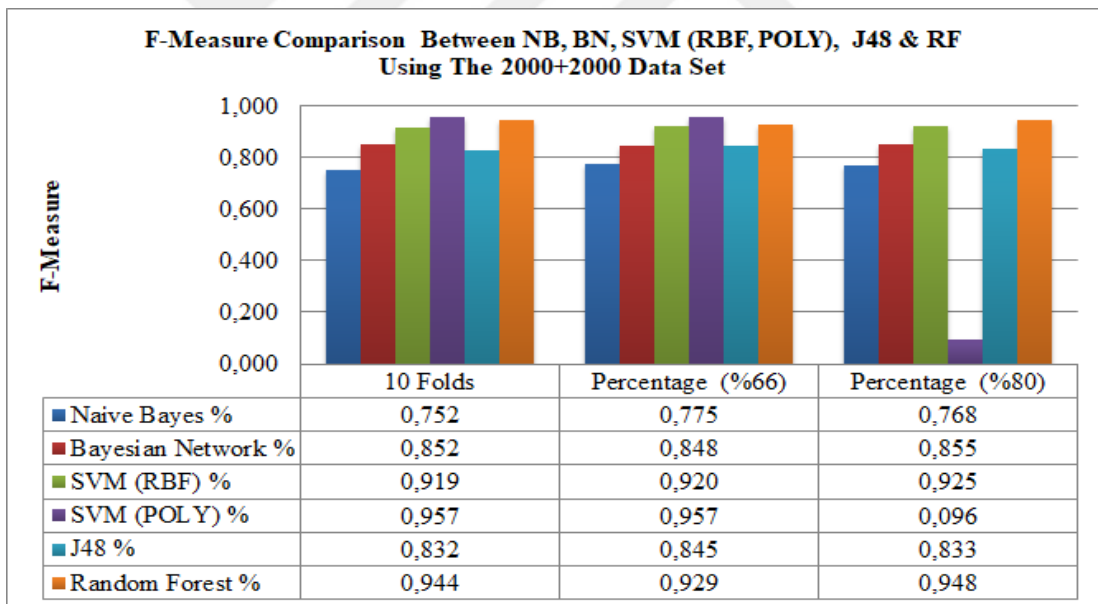


Figure 25-F-Measure comparison to 2000 ham and 2000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms

As seen on Figure 25, the best score is the Percentage %80 result of the SVM (Poly) algorithm with 2000 ham + 2000 spam data.

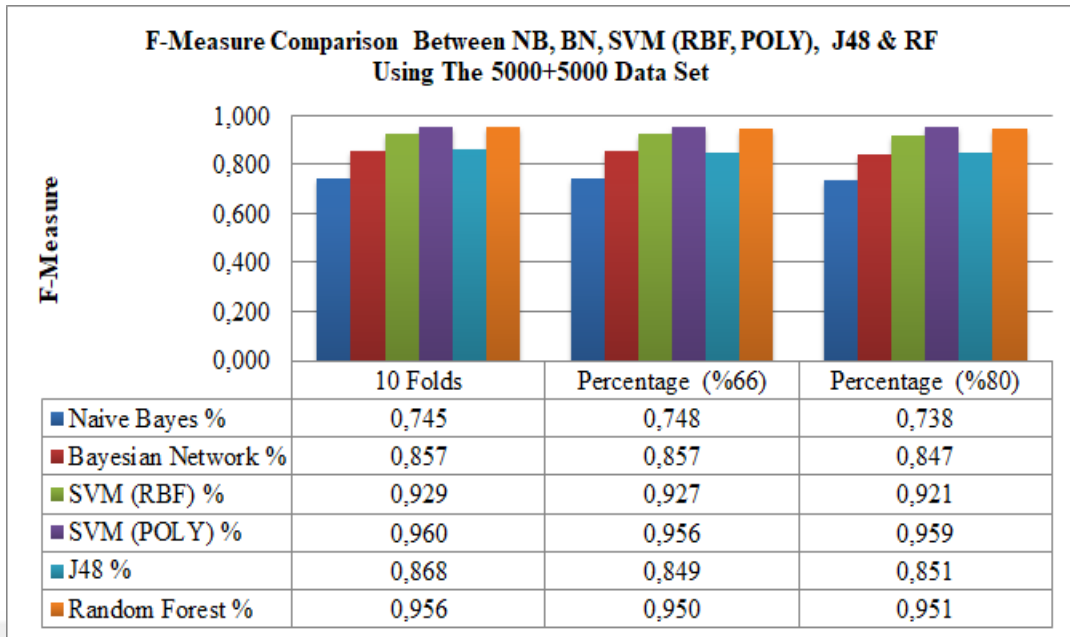


Figure 26-F-Measure comparison to 5000 ham and 5000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms

As seen on Figure 26, the best score is the 10 Folds result of the SVM (Poly) algorithm with 5000 ham + 5000 spam data.

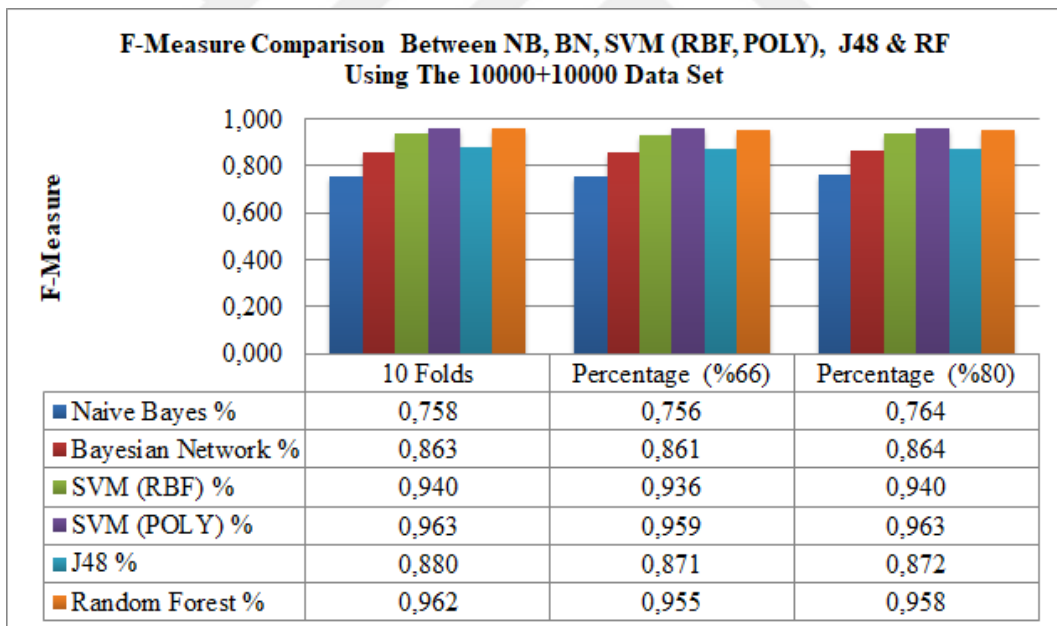


Figure 27-F-Measure comparison to 10000 ham and 10000 spam between NB, BN, SVM (RBF, POLY), and J48 & RF algorithms

As seen on Figure 27, the best score is the Percentage %80 and 10 Folds result of the SVM (Poly) algorithm with 10000 ham + 10000 spam data.

4.4 Test Results

As a result of this research, it is observed that the Random Forest algorithm is more successful in weighted TF-IDF vector representation in terms of its success results. SciKit Learn tool accuracy is high or comparable to the results of the other vector representations. The results obtained by Word2Vec are more successful using SVM (Poly) classification algorithm.



CHAPTER 5

CONCLUSION

In this thesis, the performance comparison of ML algorithms using different vector representations of emails are observed and analyzed. The methods for vectorization we used are Weighted TF_IDF, SciKit Learn based on TF-IDF and Gradient Boosting, and finally Word2Vec. We tested the best performing ML algorithms for classification using these vector representations for email classification. We applied these algorithms to different sets of data being divided into test and train data segments. We use Naive Bayes, Bayesian Network, Radial Based Function and Polynomial for Support Vector Machine, J48 tree and Random Forest algorithms. When we look at the results obtained, we have observed that the best and most successful algorithm is SVM (Poly). Based on our experimental results we conclude that Word2Vec provides better results using the vectors created by the methods used here for vector generation.

As for future study, better and more successful vector generation with big data can be applied in relation with Word2Vec vectors. Besides, using more emails (big data), better accuracy can be achieved. In addition, suitable algorithms can be selected and merged to improve the success rate of the spam detector as in ensemble learning methods.

REFERENCES

1. **Team, R. (2015)**. Email Statistics Report, 2015-2019. The Radicati Group.
2. "History of Phishing | Phishing.org", **Phishing.org, 2017**. [Online]. Available: <http://www.phishing.org/history-of-phishing/>. [Accessed: 01-Feb- 2018].
3. Available: <https://www.statista.com/markets/424/topic/1065/cyber-crime/>. [Accessed: 29- January - 2018].
4. Available:<https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/>. [Accessed: 22–January-2018].
5. **W.A, Awad & S.M, ELseuofi. (2011)**. Machine Learning Methods for Spam E-Mail Classification. International Journal of Computer Science & Information Technology. 3. 10.5121/ijcsit.2011.3112.
6. Available: <https://www.statista.com/statistics/420391/spam-email-traffic-share/>. [Accessed: 29- January- 2018].
7. **APWG (2017)**, “Phishing activity trends report”, available at: https://docs.apwg.org/reports/apwg_trends_report_h1_2017.pdf [Accessed: January 15, 2018].
8. **APWG (2017)**, “Phishing activity trends report”, available at: https://docs.apwg.org/reports/apwg_trends_report_q3_2017.pdf. [Accessed: January 15, 2018].
9. **Almomani, A., Gupta, B. B., Atawneh, S., Meulenberg, A., & Almomani, E. (2013)**. A survey of phishing email filtering techniques. IEEE communications surveys & tutorials, 15(4), 2070-2090.

10. **Kaur, G., Gurm, R. K., RIMT-IET, M. G., RIMT-IET, M. G., & Sahib, F.** A Survey on Various Classification Techniques in Email Spamming. *International Journal of Technology and Computing (IJTC)* vol, 5, 589-593.
11. **Lai, C. C., & Tsai, M. C. (2004, December).** An empirical performance comparison of machine learning methods for spam e-mail categorization. In *Hybrid Intelligent Systems, 2004. HIS'04. Fourth International Conference on* (pp. 44-48). IEEE.
12. **Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011).** Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 12(Oct), 2825-2830.
13. **Goldberg, Y., & Levy, O. (2014).** word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.
14. **Jain, G.** A Study of Bayesian Classifiers Detecting Gratuitous Email Spamming.
15. **Abu-Nimeh, S., Nappa, D., Wang, X., & Nair, S. (2007, October).** A comparison of machine learning techniques for phishing detection. In *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit* (pp. 60-69). ACM.
16. **Shams, R., & Mercer, R. E. (2016).** Supervised classification of spam emails with natural language stylometry. *Neural Computing and Applications*, 27(8), 2315-2331.
17. **Sen, D., Das, C., & Chakraborty, S.** A New Machine Learning based Approach for Text Spam Filtering Technique.
18. **Al Sarhan, A., Jabri, R., & Sharieh, A. (2017).** Website Phishing Detection Using Dom-Tree Structure and Cant-MinerPB Algorithm. *American Journal of Computer Science and Information Engineering*, 4(4), 38-42.
19. **Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Sakkis, G., Spyropoulos, C. D., & Stamatopoulos, P. (2000).** Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach. *arXiv preprint cs/0009009*.

20. **W.A, Awad & S.M, ELseuofi. (2011).** Machine Learning Methods for Spam E-Mail Classification. *International Journal of Computer Science & Information Technology*. 3. 10.5121/ijcsit.2011.3112.
21. **Chandrasekaran, M., Narayanan, K., & Upadhyaya, S. (2006, June).** Phishing email detection based on structural properties. In *NYS Cyber Security Conference (Vol. 3)*.
22. **Parsons, K., McCormac, A., Pattinson, M., Butavicius, M., & Jerram, C. (2015).** The design of phishing studies: Challenges for researchers. *Computers & Security*, 52, 194-206.
23. **Adewumi, O. A., & Akinyelu, A. A. (2016).** A hybrid firefly and support vector machine classifier for phishing email detection. *Kybernetes*, 45(6), 977-994.
24. **Moghimi, M., & Varjani, A. Y. (2016).** New rule-based phishing detection method. *Expert systems with applications*, 53, 231-242.
25. **Agarwal, D. K., & Kumar, R. (2016).** Spam Filtering using SVM with different Kernel Functions. *International Journal of Computer Applications*, 136(5).
26. **Altaher, A. (2017).** Phishing websites classification using hybrid svm and knn approach. *Int J Adv Comput Sc*, 421, 8.
27. **Tang, Y. (2013).** Deep learning using linear support vector machines. arXiv preprint arXiv:1306.0239.
28. **Hashemi, S. M.** Detection and Filtering Spam using Feature Selection and Learning Machine Methods.
29. **Zareapoor, M., & Seeja, K. R. (2015).** Feature extraction or feature selection for text classification: A case study on phishing email detection. *International Journal of Information Engineering and Electronic Business*, 7(2), 60.
30. **Diale, M., Van Der Walt, C., Celik, T., & Modupe, A. (2016, November).** Feature selection and support vector machine hyper-parameter optimisation for spam detection. In *Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech)*, 2016(pp. 1-7). IEEE.

31. **Smadi, S., Aslam, N., Zhang, L., Alasem, R., & Hossain, M. A. (2015, December)**. Detection of phishing emails using data mining algorithms. In Software, Knowledge, Information Management and Applications (SKIMA), 2015 9th International Conference on (pp. 1-8). IEEE.
32. **Wang, J., Li, Y., & Rao, H. R. (2016)**. Overconfidence in phishing email detection. *Journal of the Association for Information Systems*, 17(11), 759.
33. **Ramos, J. (2003, December)**. Using tf-idf to determine word relevance in document queries. In Proceedings of the first instructional conference on machine learning (Vol. 242, pp. 133-142).
34. **Duda, R. O., Hart, P. E., & Stork, D. G. (2012)**. Pattern classification. John Wiley & Sons.
35. **Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013)**. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).
36. **Goldberg, Y., & Levy, O. (2014)**. word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method. arXiv preprint arXiv:1402.3722.
37. Available: <https://code.google.com/archive/p/word2vec/>. [Accessed: 03- Feb- 2018].
38. **Patil, T. R., & Sherekar, S. S. (2013)**. Performance analysis of Naive Bayes and J48 classification algorithm for data classification. *International Journal of Computer Science and Applications*, 6(2), 256-261.
39. **Tsamardinos, I., Brown, L. E., & Aliferis, C. F. (2006)**. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine learning*, 65(1), 31-78.
40. **Friedman, N., Nachman, I., & Peér, D. (1999, July)**. Learning bayesian network structure from massive datasets: the «sparse candidate «algorithm. In Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence (pp. 206-215). Morgan Kaufmann Publishers Inc.
41. **Cortes, C., & Vapnik, V. (1995)**. Support-vector networks. *Machine learning*, 20(3), 273-297.

42. **Gunn, S. R. (1998).** Support vector machines for classification and regression. ISIS technical report, 14(1), 5-16.
43. **Zeng, Z. Q., Yu, H. B., Xu, H. R., Xie, Y. Q., & Gao, J. (2008, November).** Fast training support vector machines using parallel sequential minimal optimization. In Intelligent System and Knowledge Engineering, 2008. ISKE 2008. 3rd International Conference on (Vol. 1, pp. 997-1001). IEEE.
44. Available:<http://www.wiki-zero.com/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvRmlsZTpLZXJuZWxfTWfjaGluZS5zdmc>. [Accessed: 19- January- 2018].
18:07, 8 Subat 2017.
45. **George Dimitoglou, James A. Adams, and Carol M. Jim,**” Comparison of the C4.5 and a Naïve Bayes Classifier for the Prediction of Lung Cancer Survivability”
46. **Koprinska, I., Poon, J., Clark, J., & Chan, J. (2007).** Learning to classify e-mail. Information Sciences, 177(10), 2167-2187.

APPENDICES A

CURRICULUM VITAE



PERSONAL INFORMATION

Surname, Name: Nazlı NAZLI

Nationality: Turkish (TC)

Date and Place of Birth: 03.05.1989

Marital Status: Single

Phone: +90 555 807 01 61

Email: nazlinazli58@gmail.com

EDUCATION

Degree	Institution	Year of Graduation
MSc	Çankaya Univ., Computer Engineering	2014-2018
B.Sc.	Çankaya Univ., Computer Engineering	2009-2014
High School	Tuzluca YDA High School	2003-2007

PUBLICATIONS

1. **Goker, O., Nazli, N., Dogdu, E., Choupani R., Erol, M.M., (2018).** A Robust Watermarking Scheme Over Quadrant Medical Image in Discrete Wavelet Transform Domain. International Conference on Control, Decision and Information Technologies 2018 (CODIT'18).
2. **Hassanpour R., Dogdu E, Choupani R, Goker O, Nazli N, (2018).** Phishing Email Detection by Using Deep Learning Algorithms. Poster, supplemental material(s). ACM SE '18: Southeast Conference Proceedings.

WORK EXPERIENCE

Year	Place	Enrollment
2017-2017	Fatum	Software Support
2015	Diva Soft	Senior Engineer
2011-2014	Çankaya University	Part-Time St.
2012 July	TETAŞ	Intern
2012 June	BAŞARSOFT	Intern
2011 June	BAŞARSOFT	Intern

PROJECTS

1. Corasis (ERP)
2. Task Management System (ASP.NET)
3. CBS OpenLayers Web Project, CBS Project To Desktop Application (BaşarSoft MapInfo)
4. Alumni Portal (SharePoint) (Senior Project)

FOREIGN LANGUAGES

English	: Intermediate oral and writing, reading skills
Italian	: Beginner
Spanish	: Beginner
Ottoman Turkish	: Elementary
German	: Beginner

MEMBERSHIPS IN PROFESSIONAL ORGANIZATIONS

1. Member of Çankaya University Computer Engineering, Information Technology and Artificial Intelligence Clubs
2. Chorus of Turkish folk music and classical music
3. Robotic and Artificial Intelligence Community

HOBBIES

Writing, travelling, photography, painting

