

1-1-2010

Spam detection system: a new approach based on interval type-2 fuzzy sets

Reza Ariaeinejad
Ryerson University

Follow this and additional works at: <http://digitalcommons.ryerson.ca/dissertations>

 Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Ariaeinejad, Reza, "Spam detection system: a new approach based on interval type-2 fuzzy sets" (2010). *Theses and dissertations*. Paper 986.

Spam Detection System:

A New Approach Based on Interval Type-2 Fuzzy Sets

By:

Reza Ariaeinejad

B.Sc. Computer Engineering

Islamic Azad University, Iran, 2004

A Thesis presented to Ryerson University

In partial fulfillment of the degree of

Master of Applied Science

In the program of

Electrical and Computer Engineering

Toronto, Ontario, Canada, 2010

© Reza Ariaeinejad

Author's Declaration

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Reza Ariaeinejad

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, at the request of other institutions or individuals for the purpose of scholarly research.

Reza Ariaeinejad

Borrower's Page

Ryerson University requires the signatures of all persons using or photocopying this thesis.
Please sign below, and give contact info and date.

Spam Detection System: A New Approach Based on Interval Type-2 Fuzzy Sets

Reza Ariaeinejad

Master of Applied Science

Department of Electrical and Computer Engineering

Ryerson University, Toronto, ON, Canada, 2010

Abstract

Today, most Internet users use email to communicate electronically. They depend on the Internet to deliver their important emails safely and to the right recipients. However, the fast growth of Internet users and their use of email together with the exponential increase of unsolicited users sending spam have made the email system less reliable. An email can falsely be marked by a spam filter on its way to the recipient or even get buried among junk mail in the recipient's inbox. There are several intelligent anti-spam filters which use different artificial intelligence methods to detect spam including neural networks and fuzzy logic systems. This paper presents an interval based type-2 fuzzy spam detection system. Our results show that interval type-2 fuzzy set is an effective technique for spam detection and email classification. The proposed system enables the user to have more control over the various categories of spam and allows for filter personalization.

Acknowledgements

First of all, I would like to thank my family who has always been supportive and encouraging in the most critical moments throughout my life.

Also, I would like to thank Dr. Alireza Sadeghian for his direction, supervision and invaluable advice throughout this project. Without his precious guidance, help and advice, it was impossible to finish this thesis successfully.

Also, I would like to thank Dr. Hooman Tahayori for sharing his novel idea of using the interval type-2 fuzzy sets in spam detection with me. I would also like to thank him for his precious help during the different stages of the project.

Lastly, I would like to thank my dear friends who helped me stay balanced during the stressful times.

Table of Contents

Title Page	i
Author’s Declaration.....	ii
Borrower’s Page.....	iii
Abstract	iv
Acknowledgements.....	v
Table of Contents	vi
List of Tables	ix
List of Figures.....	x
List of Formulas	xi
1. Chapter 1: Introduction and Problem Definition.....	1
1.1. Spam Definition and History	1
1.2. Existing Spam-Filtering Methods.....	2
1.3. Motivation	3
1.4. Objectives	3
1.5. Scope of the Study	4
1.6. Thesis Structure	5
2. Chapter 2: Background Info and Literature Review	6
2.1. Internet Messaging/Mailing System	6
2.2. The Process of Sending an Email	6
2.3. Email Security Measures	7
2.4. What is Spam?	7
2.5. Spam Categories	8
2.6. Advanced Spamming Techniques	9
2.7. Anti-Spam Legislation Efforts.....	9
2.8. Popular Spam-Filtering Methods.....	10
2.8.1. Gray-listing.....	11

2.8.2.	Sender Policy Framework	11
2.8.3.	Domain-keys	11
2.8.4.	Real-time Black Lists.....	12
2.8.5.	Spam-assassin	12
2.8.6.	Learning-Based Spam-Filtering Methods	13
2.8.7.	Fuzzy Logic Based Spam-filtering Systems	14
2.9.	Sets: Description and Formalization	16
2.10.	Intervals.....	17
2.11.	Fuzzy Sets.....	17
2.11.1.	Description of the Fuzzy Sets Concept.....	17
2.11.2.	Fuzzy Sets Formalization	18
2.11.3.	Main Classes of Membership Functions.....	19
2.12.	Type-2 Fuzzy Sets and Intervals.....	21
2.13.	Centroid of a Fuzzy Set.....	23
3.	Chapter 3: Methodology and Experimental Procedure	25
3.1.	System Design Scheme	25
3.2.	Building the Dictionaries	25
3.2.1.	Weight Calculation	27
3.3.	Parsing.....	29
3.4.	Checking with all Dictionaries.....	30
3.5.	Building Fuzzy Maps	32
3.5.1.	Distance Calculation.....	33
3.5.2.	Weight of Each Interval in the Map (Frequency of each Word)	34
3.6.	Evaluation and Prediction	36
3.7.	Dictionary Improvement	39
3.8.	Results and Analysis	40
4.	Chapter 4: Conclusion and Future Work.....	48
4.1.	Conclusion	48
4.2.	Future Work	50

Appendix I. The Unusual Signs in the Words	51
Appendix II. White Dictionary Words List.....	52
Appendix III. Adult Dictionary Word List	53
Appendix IV. Other Dictionaries.....	54
References.....	55

List of Tables

Table 1. Contingency Table.....	53
Table 2. Main System Results	54
Table 3. Hotmail Contingency Table.....	56
Table 4. Hotmail Results.....	57
Table 5. Yahoo Contingency Table.....	57
Table 6. Yahoo Results.....	58
Table 7. Computer Science Data Set Results.....	59

List of Figures

Figure 1. Fuzzy Set A and its Height, Core and Support.....	30
Figure 2. Overall System Architecture.....	37
Figure 3. A Sample of the 3D Fuzzy Map.....	45
Figure 4. A Schematic of the Fuzzy Subsets	49
Figure 5. Hotmail Snapshot.....	56
Figure 6. Yahoo Snapshot.....	58

List of Formulas

1. Interval Width	29
2. Interval Center	29
3. Fuzzy Set.....	29
4. Fuzzy Supplement.....	31
5. Fuzzy Core.....	31
6. Triangular Fuzzy Set.....	31
7. Triangular Class Parameters	31
8. Trapezoidal Fuzzy Set.....	32
9. Gaussian Fuzzy Set.....	32
10. Non-Symmetric Gaussian Fuzzy Set.....	32
11. Parabolic Fuzzy Set.....	32
12. Fuzzy Type-2 Set.....	33
13. Interval Fuzzy Type-2 Set.....	34
14. FOU of a Fuzzy Set	34
15. UMF of a Fuzzy Set.....	35
16. LMF of a Fuzzy Set.....	35
17. J of a Fuzzy Set.....	35
18. FOU of a Fuzzy Set.....	35
19. Centroid of LMF.....	35
20. Centroid of UMF.....	35
21. C-left High.....	36
22. C-right Low.....	36
23. C-left Low.....	36
24. C-right High.....	36
25. Union Formula for Dictionary Build-up.....	38
26. Weight Formula.....	39
27. Expected Value Formula.....	40
28. Standard Deviation.....	40
29. Standard Deviation (Another Way).....	40

30. Jaro Distance.....	43
31. Matching Formula.....	43
32. Transposition Formula.....	43
33. Jaro-Winkler Distance.....	43
34. Term Frequency.....	46
35. Inverse Document Frequency.....	47
36. TF-IDF Formula.....	47
37. Fuzzy C-mean.....	50
38. Degree of Membership.....	51
39. N-dimension Center of the Cluster.....	51
40. Spam Accuracy.....	53
41. Spam Precision.....	54
42. Spam Recall.....	54

1. Chapter 1: Introduction and Problem Definition

1.1. Spam Definition and History

Internet is one of the most popular forms of media consumed by our society. The majority of Internet users rely on email to communicate electronically and they depend on the Internet to safely deliver their email to the right recipient. There are millions of emails sent and received every day. Among those, there are some unwanted emails known as “Spam”, an expression that originated from a Monty Python sketch [1].

Today, spam refers to junk, trash or unwanted email. The opposite of spam is referred to as “Ham”, which is a genuine or desirable email. Spam is generated for many reasons, such as selling a product, acquiring personal information from users, spreading viruses and worms, advertising, political advocacy, etc.

Regardless of the reasons for sending these junk emails, they create unnecessary traffic on the networks, impose unnecessary expenses on our resources and make the emailing system unreliable because of the imperfect nature of spam-filtering systems. A genuine email can falsely get caught by spam filters before it gets to the right recipient or it may be misplaced among junk email in the user’s inbox.

It is estimated that spam costs each US email user \$30-50 annually in lost time and costs each employee \$730 annually reduced productivity [2, 62]. Compounding those losses, it is further estimated that US companies lose \$8,900,000,000 per year as a result of the spam issue. Given these numbers, it is clear that corporations and individuals that use electronic mail and the Internet would save large amounts of time, money and resources if they could avoid spam. The same is also true for Internet Service Providers, or ISPs, and Email Service Provider, or EMPs, if the problem could be solved or at least reduced. One of the most important goals of EMPs is to identify and filter the unwanted spam and to make the server more usable. Most Internet users have experienced some form of spam and are able to distinguish between it and ham. However, the first researcher who officially wrote a request for comments in 1974 was Joe Postel [3]. The spam issue has been growing ever since.

1.2. Existing Spam-Filtering Methods

There are number of measures to limit and prevent spam. These include user awareness, technological solutions (such as spam filtering) and even through legal action.

From a technological standpoint, there are software-based tools which detect and block spam automatically. These tools are called spam-filters. Since spammers use a variety of established techniques to penetrate users' inboxes, such as using fake addresses, there are several filters used to block these attempts, from blacklists to content-based filters. Content-based filters have been proven to be more powerful than blacklists. There are three major categories of content-based spam filters: Learning-based, Rule-based and Keyword-based.

The Keyword-based filter [30] uses a dictionary of commonly used spam words and searches for similar words in the text or document. This filter requires constant maintenance. Rule-based filters benefit from a vast range of categorized tests that find junk mail characteristics. It assigns a unique "score" to each email and then decides if an email is spam based on that score [31, 32]. Although Rule-based filters work well, they require periodic maintenance and update also, since their rules are fixed and become outdated over time. Learning-based filters have the benefits of the two aforementioned techniques with an important advantage over both. These filters use machine learning techniques to update automatically and do not require periodic manual updates [56].

In addition, here is a list of currently existing non content-based methods of spam-filtering, which will be explained in detail in the second chapter:

- Gray-listing
- Sender Policy Framework
- Domain-keys
- Real-time Black Lists
- Spam-assassin

Some of these non content-based methods also benefit from the use of machine learning techniques. However, the main focus in this thesis is content-based spam filters and specifically on the Learning-based subcategory.

There are many different machine learning techniques that have the potential to be used in the learning-based spam filters. Techniques such as Boosting Tree [31], Support Vector Machine [34], Decision Tree [35], K-nearest Neighbor [34] and Fuzzy Logic [56 -61] have already been employed for this purpose with promising results. However, the first technique that introduced a spam filter based on machine learning techniques was Sahami et al. [33]. He proposed a Naïve Bayesian classifier trained on the previously detected spam and ham emails to categorize unseen messages. It performed well on unseen messages, which led to the rapid increase of the use of machine learning techniques in spam detection.

1.3. Motivation

Although the existing spam detection methods have shown to be effective and reliable, there is always room for improvement. Some methods require periodic maintenance. Other methods, which have solved for those weaknesses, are less effective at filtering spam from ham. The motivation for this thesis is to improve the existing methods in the sense of results. Furthermore, we shall propose a method that functions autonomously, without the need for periodic updates. Also, this proposed method must prove to be more efficient and must yield promising results.

1.4. Objectives

In this thesis, we employ interval type-2 fuzzy sets to build an automatic spam filter. Type-2 fuzzy sets are a generalization of type-1 fuzzy sets so they can cope with more uncertainty [37]. From their inception, there have been extensive discussions in fuzzy logic and in the greater fuzzy sets community regarding the problem of membership functions of type-1 fuzzy sets, which do not have the associated capability to deal with uncertainty. This seems to contradict arguments related to the need for fuzzy sets. Professor Zadeh [38], the father of fuzzy sets, resolved the problem by proposing more sophisticated forms of fuzzy sets. The first category of

these sets is called a “type-2 fuzzy set”. Type-2 fuzzy sets are an attempt to resolve the issue associated with the type-1 fuzzy sets by incorporating the notion of uncertainty about their membership functions into fuzzy set theory.

Our proposed approach is to use interval based type-2 fuzzy sets to classify an email either as ham or as spam.

Furthermore, most anti-spam filters count the number of spam words and compare that with the number of legitimate words in the message to decide whether or not it is spam [33, 39]. Therefore, smart spammers have devised a new technique, which inserts random, meaningless text into the email to offset that percentage. As a result, the message is considered to be deliverable [40]. However, in our system we try to approach this problem in a systematic way using fuzzy maps that allow us to decide if the message is ham or spam.

The proposed system also has an adaptive nature that is if the spam trends change over time with spammers employing new techniques such as using new wording or using spaces between the word’s letters, etc. our system can easily adapt and self-update thereby coping with latest spam techniques.

1.5. Scope of the Study

The main objective of this study is to employ an interval fuzzy type-2 system to classify emails as either spam or ham and also categorize spam emails into the right groups. The scope of this study is limited to the emails that contain plain text only. Our design is restricted from processing links, pictures or any other material in the email but plain text only.

However, we have the ability to deal with the basic spamming techniques [see section 2.6.] such as: Scrambled Text, which is leaving spaces or other signs among the word’s letters; Invisible Text, which is inserting words to neutralize statistical analysis software and always use the same colour as background for the text; Letter Randomization, which are long, useless strings designed to neutralize signature-based filters, and Character Set Encoding, which usually uses base64 and printable character set encodings to “hide” words from clear format of the text.

Moreover, if spam trends change over time, our system has the flexibility to adapt to the new words [see section 3.8.]. That is to say it is self-adaptive.

Additionally, we have used a technique that extracts the roots of the words. For instance, we eliminate the “ing” or “ed” from the end of the words and extract the root of each word or at least get as close as possible to the root. Using this technique will eventually help us to deal with the other languages that have similar roots with English language.

1.6. Thesis Structure

The remainder of this thesis is organized as follows. Chapter two provides the background information and literature review to give the reader as much information as is necessary to fully understand the problem, the existing methods, and also all the information that may be needed to understand the way our system implemented and works. Chapter three is the methodology and experimental procedure. Here we introduce our proposed method and explain how it has been implemented. This chapter concludes with our results. Our conclusion is articulated in chapter four. Here we summarize the whole work and provide our perspective on future research.

2. Chapter 2: Background Info and Literature Review

2.1. Internet Messaging/Mailing System

The Internet mailing system Simple Mail Transfer Protocol, or SMTP has been the main protocol for sending email on Internet for several years is described, in detail, by RFC 2821[4].. The first SMTP appeared in 1981. This has since been superseded by newer generations of SMTP RFCs, which are backward compatible with some new functionality added.

An email consists of a body and a header. Headers have different fields and are fully described in RFC 2822 [5]. The date header describes the time and date that the email was finished and sent out. “From” and “reply-to” fields describe the sender’s email address and the destination address of a reply. The recipient’s address goes in the “To” field. The Carbon Copy, or CC, and Blind Carbon Copy, or “BCC”, fields can also specify the email recipient. The content of the “To” and the “CC” fields can be seen by recipient(s), whereas the content of the “BCC” field cannot be seen by recipient(s). The “Message-ID” field specifies the ID of the email and the “References” and “In-Reply-To” fields show if the email is a reply to a previously sent email. The body of the email contains the actual message that author of the email has provided [62].

2.2. The Process of Sending an Email

As soon as the author has finished the body and header of the email, the computer would try to connect to a SMTP server, which is hosted at the senders ISP. This process is similar to sending a letter by a post-person. The email travels among many SMTP servers until it reaches the right server, which will put the email in the recipient’s inbox. The first SMTP server looks up the recipient’s address in the Domain Naming System, or DNS. DNS functions like a telephone book. It translates the actual server names to their equivalent Internet Protocol, or IP addresses. An IP address is a numerical label that is assigned to any computer or device. The address acts as a node in a network that uses the Internet Protocol for communication between these mentioned nodes. The look up process will return the IP address of the first and closest SMTP server that

the email should be sent to. This can be the actual email recipient or another intermediary SMTP server [62].

2.3. Email Security Measures

According to [insert the name of the professor or the paper that you are referencing], in order to have a secure and accessible email system, we need to be able to support three different essential security measures [19]:

- **Confidentiality:** Protecting emails and computers from unauthorized or unknown access
- **Integrity:** Guaranteeing that emails and computers are not destroyed or distorted through an unauthorized access.
- **Availability:** Ensuring that email servers meet the reasonable service level.

2.4. What is Spam?

There are millions of emails sent and received every day. Among those, there are some unwanted emails that called “Spam”. Spam is an expression that was coined in a Monty Python sketch [1]. Today, spam refers to junk, trash or unwanted email. There are many different reasons for sending spam such as selling a product, acquiring personal information from users, spreading viruses and worms, advertising, political advocacy, etc. The opposite of spam, which is a genuine or desirable email, is referred to as “Ham”.

2.5. Spam Categories

According to [insert the name of the person or paper that you are referencing], there are ten main categories of spam [46]. They are:

- 1- **Adult:** primarily consists of content for mature audiences.
- 2- **Financial:** primarily consists of financially fraudulent content.
- 3- **Fraud:** contains any sort of fraud other than financial fraud.
- 4- **Health:** primarily contains content regarding pharmaceutical goods and products.
- 5- **Internet:** primarily contains advertisements.
- 6- **Leisure:** primarily contains marketing content devoted to selling leisure goods and services.
- 7- **419-spam:** contains content that seeks to solicit a monetary sum from the recipient by guaranteeing further monetary gain following the initial cash advance.
- 8- **Political:** contains content pertaining to political campaigns.
- 9- **Products:** contains promotional or marketing content for products outside of the health or leisure categories.
- 10- **Scams:** Contains any scam not identified in the categories above.

2.6. Advanced Spamming Techniques

Fifteen years ago spammers did not have to think about anti-spam techniques for they did not yet exist. Over time, with the high rate of increase in spamming organizations were forced to think more about this issue and to deploy effective spam protection techniques. These new techniques, in turn, led to the development of new sending techniques, which allowed spammers to avoid having their spam be detected [19]. A few of those sending techniques are:

- **Scrambles Text: Breaking up a word by** inserting spaces or other characters among the letters of spam words to break a word. For instance, the word “capital” could be written as “c a p i t a l” or “c-a-p-i-t-a-l” or “c*a*p*i*t*a*l”.
- **Invisible Text:** Inserting additional, irrelevant words in order to neutralize statistical analysis software while using the same colour as the background for the text. Typically this shared colour will be white, which hides the irrelevant text from the user when the email is rendered by the client machine.
- **Split Words:** Splitting or interrupting spam words such as “Lover” or “Girls” by inserting HTML Tags into them.
- **Letter Randomization:** Inserting long passages of irrelevant text into the body of the email in order to confuse signature-based filters.
- **Character Set Encoding:** Using base64 and printable character set encodings in order to “hide” spam words from the clear text format.

2.7. Anti-Spam Legislation Efforts

Due to the considerable amount of financial loss generated by spam as well as a lack of existing legal measures to prevent sending spam, new laws and legislations were passed to address this problem. The United States, Canada and the European Union have all taken legal action to

respond to this issue. In 2002, the European Parliament passed the European Union Privacy and Electronic Communications Directives (EUPECD), 2002/58/EC. In 2003, the USCAN-SPAM Act, also known as the Controlling the Assault of Non-Solicited Pornography and Marketing was passed. These are both legal efforts to respond to the spam issue.

The EUPECD prohibits unsolicited commercial or marketing communication except when the sender has obtained the prior consent of the recipient [17]. The USCAN-SPAM Act only permits unsolicited emails that adhere to a set of restrictions. For example, deceptive subject lines are forbidden and senders must clearly mark the emails as advertisements. A legitimate return email address, a physical address of the sender and an opt-out link must also be included in unsolicited emails. The USCAN-SPAM Act prohibits falsifying header information and illegally using captured third party computers to relay messages [22].

2.8. Popular Spam-Filtering Methods

There are several methods used to destroy or control spam messages [8]. One method is to block all insecure outgoing email sessions from the recipient system and have all email hosts within the network use a secure server when attempting to send emails. This method prevents abusive third parties from using the network's resources to relay emails.

Another method is to detect and filter the spam at the recipient's computer. Spam can either be stopped before it enters the receiver's computer or afterward. The former is carried out by issuing either a temporary or permanent error code in the message delivery sessions. If it is a temporary code, an original compliant sender system will eventually try to send the email. If it is a permanent error code, the email will be stopped forever. If for any reason, the email could not be delivered to the right recipient, a warning message would be sent to the sender. However, when the process is successful and the recipient receives the email, the sender will not be notified even if the recipient's system uses a spam filter and categorizes the email as spam. As such, using an internal filter is more likely to increase the risk of losing information.

In addition to the content-based filters described above, the following are several methods of non content-based spam detection [62].

2.8.1. Gray-listing

Gray-listing has been developed with the goal of having minimal impact on users. It also requires minimal maintenance [9]. It uses three pieces of information, which is commonly referred to as a triplet. A triplet is the host's IP address that attempts the delivery, the address of the sender and the address of the recipient. If the filter does not recognize a triplet, it simply denies the delivery for a certain period of time. After that certain period of time, the triplet is no longer unrecognized. The triplet becomes familiar and the messages pass through. However, gray-listing usually works on spammers that send spam but who do not examine the error codes that SMTP generates.

2.8.2. Sender Policy Framework

Sender Policy Frameworks, or SPF, have been adopted by most of large email service providers such as Yahoo, Google and Hotmail [10]. It prevents forging arbitrary email addresses as an envelope sender by spammers in SMTP. It forces administrators to publish the IP addresses that are permitted to send an email, which is done at the DNS level. SPF also checks all incoming emails to see if they come from a permitted address. If email comes from a forbidden address, it is considered spam. SPF is most effective when combined with other spam-filtering techniques.

2.8.3. Domain-keys

The Domain-keys technique is also used by most major email service providers [11]. This technique attempts to detect if the header of an email has been changed after sending and prevents spam from forging arbitrary sender email addresses. With this technique, all senders must sign outgoing emails and the system must generate a pair of keys, one public and one private. The private key allows the sender system to sign outgoing emails and the public key is made available to the DNS server. When the email recipient checks these two keys, they must be identical or system considers the email to be spam.

2.8.4. Real-time Black Lists

Real-time Black Lists, or RBL, are usually distributed via DNS servers [12]. They contain the IP addresses of the hosts which are suspected of being insecure. As the name suggests, the lists are updated in real-time. RBLs work at the network level. If a server's IP is added to the list, all the users of that sever will be blocked. Also, it assigns degrees to each IP address in the list so some addresses are blocked instantly and others are marked as suspicious. RBL's have several rules it applies in order to consider a server as a potential threat, such as open proxies, open mail servers, etc. The major drawback of this method is a large volume of false spam ratio.

2.8.5. Spam-assassin

Spam-assassin is a rule-based spam filtering system with an extensive built in set of rules [12]. The set of rules that spam-assassin has, is basically a variety of mechanisms used to identify spam from simple rules such as looking for a missing subject line to more complicated mechanisms including: Bayesian filtering, White and Black lists, DNS block-lists, network-based clearing address lists and collaborative filtering databases such as Pyzor, DCC, etc.

After receiving an email, it will be checked against the rule set and if it matches the rules, then a certain "score" (number of points) associated to that specific rule would be added to the checksum of the email. The total sum will be checked against a user defined or default threshold. If it is equal to or higher than that threshold, a warning message will be added to the email stating its likelihood of being a spam. In the newer versions of spam-assassin, there are two thresholds to be defined: one is for spam and the other is for possible spam.

Spam-assassin is not completely flexible because the rules and scores are static for a specific version. However, the strength-point of spam-assassin is that it can take advantage of getting input from other methods, such as razor.

Learning-Based Spam-Filtering Methods

A number of machine learning classification techniques have already been proposed for spam filtering applications. Based on [25], the following characteristics of spam filtering tasks may cause data mining issues:

- **Suddenly changing class distribution:** the amount of ham and spam changes significantly over time.
- **Unequal and uncertain error costs:** the cost of losing a legitimate message may not be equal to that of receiving a junk email.
- **Disjunctive and changing target concept:** the ability of spammers to develop new spamming techniques on a regular basis.
- **Intelligent adaptive adversaries:** Spam types and trends change over time.

The need for sufficient amount of training data is yet another complicating factor. In [26] a technique called co-training was proposed to overcome these problems. It allows the system to be trained by a small portion of labeled data. This data is used for the systems' initial training. From that point on, that system is trained with a larger amount of data which is unlabeled. The data is eventually labeled by the system and is used in an iterative process to improve the system.

With any spam filtering technique, two types of errors always occur to some degree:

- Wrongly classifying spam as ham
- Wrongly classifying ham as spam

Wrongly classifying spam as a legitimate email will likely just inconvenience the user. However, classifying ham as spam could cause more negative consequences, such as the loss of important and valuable information.

The solution may be the use of game theory [27]. It could also be one of the two other techniques proposed by Yih et al., which have low false positive rates. However, different types of users have different expectations. In a military situation, information could be vital and therefore the loss of information could have serious consequences, whereas in a personal scenario, the loss of information would likely be less harmful. Therefore, the most reasonable solution is to consider the cost of the two types of errors as a user-defined parameter [28].

The first researcher who introduced a spam filter based on machine learning techniques was Sahami et al. [33]. He proposed a Naïve Bayesian classifier which is trained on the previously detected spam and ham emails to categorize unseen messages. It performed well on unseen messages. It also resulted in a rapid increase of using machine learning techniques in spam detection. However, there are many different machine learning techniques that have the potential to be used in the learning-based spam filters. Techniques such as Boosting Tree [31], Support Vector Machine [34], Decision Tree [35], K-nearest Neighbor [34] and Fuzzy Logic [56 -61] have already been employed for this purpose with promising results.

2.8.6. Fuzzy Logic Based Spam-filtering Systems

The results of clearly prove that fuzzy logic is an excellent method for spam detection [56 – 60]. The use of linguistic variables and approximate reasoning makes fuzzy logic an ideal technique to model a problem and arrive at a useful answer. Below is a summary of the work of several researchers that effectively utilized fuzzy logic methods in the field of spam detection.

Fuad et al. introduced a trainable fuzzy type-1 spam-filter [56]. The filter has a learning period during which it could develop an effective rule-set and then apply the rule set to classify unseen messages. The other positive aspect of their filter is that it was not limited to email text. They increased the efficiency of their filter by considering and extracting other aspects of the email from its header, such as an empty sender field, etc.

Kim et al. presented a fuzzy type-1 inference approach as a feature selection method and performed a comparative experiment on the Adult spam category [57]. Their system's performance improvement, in terms of spam accuracy, was not extremely significant compared to conventional spam-filtering systems. However, other factors, such as the average error rate, spam precision and spam recall, were approximately 6-10% higher than those of conventional systems.

El-Alfy et al. [58] introduced an interesting fuzzy similarity-based spam-filtering method. The method considered the similarity of the content of the message to predict the category of spam instead of relying on a fixed pre-specified set of keywords. It therefore had the advantage of partially adapting to the new techniques and tactics that spammers could develop. El-Alfy et al. created a sort of knowledge-base to keep track of these spamming tricks dynamically. They showed that their system can provide better results compared to naïve Bayesian classifier.

Hu et al. [59] presented a spam-filtering system that interestingly was based on fuzzy clustering instead of fuzzy classification. Their system did not have any training period, nor did it take any time for training. Also, their results showed reasonably good filtering quality. The advantages of their system were: high flexibility, reduced need for manual labour, fewer privacy issues and reasonable efficiency.

Meizhen et al. [60] proposed a very interesting behavioural approach to spam detection. Instead of determining if a message is spam or ham merely based on its content, they developed a spam behavioral recognition model based on a fuzzy decision tree. Their system analyzed the information from all characteristics of emails to process them with their fuzzy decision tree. Then, by data globalization, they arrived at potential behavioural features of the emails. Although their results are not very impressive, their approach is novel and notable.

Tahayori et al. [61] introduced an interval type-2 fuzzy set methodology for the email classification. They have mathematically discussed the technique as a good method for email classification but no results have been reported since it was never implemented. Also, it seems that their approach missed a key part, which is 3rd dimension of their maps. They have also

suggested that this technique is not sufficient for a complete classification and it must be combined with other techniques.

In our proposed system, we will use interval type-2 fuzzy sets to categorize emails rather than type-1 fuzzy sets. If fully functional, our system will be capable of self-updating, through the automatic updates of its dictionaries. In other words, it is designed to be adaptive to changes in spam trends over time. We have developed a new way of calculating and assigning the weight of each word in our dictionaries, as described below in section 3.5.2. To determine if an email is spam or ham, we will develop a 3D fuzzy map, in section 3.6, for each email. Then we will categorize the email by using a data clustering technique on that map. We have tried to develop a powerful spam-filtering system that will be capable of recognizing unseen spam and adapting to potential changes in spamming techniques as efficiently as possible.

2.9. Sets: Description and Formalization

Indeed, we can show every mathematical concept with the set theory [21], from natural to real and rational numbers. Using the set theory, on the one hand, accommodates conceptual innovation and on the other hand, makes it easy to represent information of a complex nature. For understanding the set theory we need to first understand the concept of Universal Set or \mathbf{X} . Universal set represents the universe of discourse which contains any possible element that in any way relates to our purpose. An important and most usually used universal set is the set of all points in n-dimensional space which is shown as \mathbf{R} . The relationship of each element with the universal set or any other subset of the universal set is shown by the sign the sign \in (belongs to) or the sign \notin (doesn't belong to). We can define a set in two different ways. One way is to define the set itself and the other way is to define the set by its elements. Also, the total number of the elements of each set is called the cardinality of the set. And it's shown by $|A|$. The concept of a subset is shown by the sign \subset . We can say $A = B$ only if $A \subset B$ and $B \subset A$. if a set has no elements in it we call it an empty set and show it by the sign \emptyset . Also, the compliment of a set is shown by \bar{A} .

2.10. Intervals

Intervals are special cases of sets [21]. These are connected subsets if in the \mathbf{R} , which give us a good way of approximating real life situations. We show the intervals by lowercase letters in brackets, for example: $[x]$. They have lower and upper bounds that are shown by x^-, x^+ , so intervals may be closed or not. Here are the formulae to calculate the center and the width of an interval:

$$\text{Width} ([x]) = x^+ - x^- \quad (1)$$

$$\text{Center} ([x]) = (x^+ + x^-) / 2 = x^- + \text{width} ([x]) \quad (2)$$

2.11. Fuzzy Sets

2.11.1. Description of the Fuzzy Sets Concept

In the ordinary set theory we have solid boundaries, either limited or limitless, for a set. However, fuzzy sets offer a new concept of continuous boundaries. This idea is based on the human perception of processes. In life, we do not use only yes and no when we talk or describe something either a concept or an idea. Our natural language is the best illustration of the fuzzy set concept. We use words such as small, big, hot, cold, long, short, etc. which describe relative traits that do not have solid boundaries. Although we have no problem using these concepts in real life, it could make a lot of trouble when we want to express them in a set-based model. For instance, yes can represent 1, or ‘include’, and no can represent 0, or ‘exclude’, for the elements. However, we do not have anything for concepts such as high pressure or hot temperature. We need to have some sort of element that can represent these concepts [21]. Zadeh was the first researcher who came up with a solution. He invented the term Fuzzy set which admits partial membership of elements.

We define a fuzzy by its membership function $\mathbf{A}(x)$ or $\mu_A(x)$:

$$\mathbf{A}: \mathbf{X} \rightarrow [0, 1] \quad (3)$$

$A(x)$ defines the degree of membership for each single element of x in A . If $A(x) = 1$ then it means that x fully belongs to A and if $A(x) = 0$ it means that x doesn't belong to A at all. However, when $A(x)$ is between 0 and 1 it means that x partially belongs to A based on the value of $A(x)$. The larger the value assigned to $A(x)$, the stronger the degree of association.

2.11.2. Fuzzy Sets Formalization

We can fully describe the fuzzy sets by their membership functions. However, we need descriptors to help in their characterization. The descriptors of a fuzzy set are: height, core and support [21].

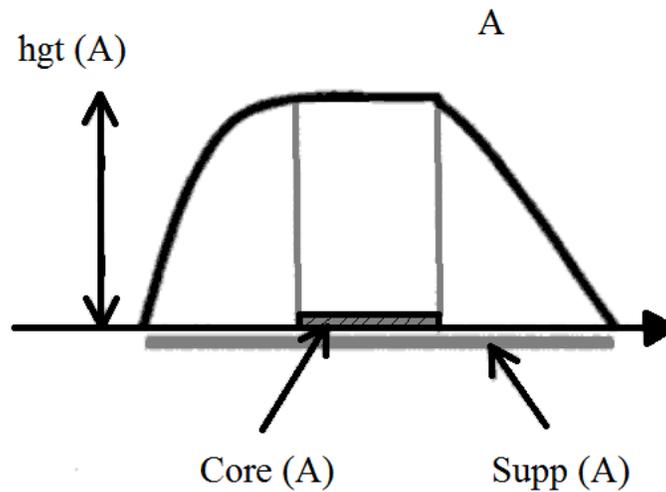


Figure 1. Fuzzy set A and its height, core and support

Hgt (A) or the height of A is the supremum of the membership function. If the $\text{hgt}(A) = 1$ then we call the fuzzy set a normal fuzzy set and in any other case, we call it subnormal. Subnormality is usually the result of dealing with a concept that has no elements that can fully satisfy it. The Core (A) is all elements which totally satisfy the fuzzy set and the Supp (A) or support of A is all elements that are partially satisfying the fuzzy set.

$$\text{Supp (A)} = \{x \in X \mid A(x) > 0\} \quad (4)$$

$$\text{Core (A)} = \{x \in X \mid A(x) = 1\} \quad (5)$$

2.11.3. Main Classes of Membership Functions

The way that a partial membership is represented depends on the concept and a good membership function should be selected based on the needed application. There are many different ways of representing the membership function based on the problem. Below, the most famous ones are listed [21].

Triangular Fuzzy Sets:

$$A(x; a, m, b) = \begin{cases} 0 & \text{if } x \leq a \\ \frac{x-a}{m-a} & \text{if } x \in [a, m] \\ 1 - \frac{b-x}{b-m} & \text{if } x \in [m, b] \\ 0 & \text{if } x \geq b \end{cases} \quad (6)$$

The parameters of the class of fuzzy sets describe the linear segments of the membership function and it could be rewritten in concise formatting with min and max functions.

$$A(x; a, m, b) = \max \{ \min [(x-a) / (m-a), (b-x) / (b-m)], 0 \} \quad (7)$$

Trapezoidal Fuzzy Sets:

$$A(x; a, m, n, b) = \begin{cases} 0 & \text{if } x \leq a \\ \frac{x-a}{m-a} & \text{if } x \in [m, b] \\ 1 & \text{if } x \in [m, n] \\ 1 - \frac{b-x}{b-m} & \text{if } x \in [n, b] \\ 0 & \text{if } x \geq b \end{cases} \quad (8)$$

Gaussian Fuzzy Sets:

$$A(x; m, \sigma) = \exp(-(x-m)^2 / \sigma^2) \quad (9)$$

Non-symmetric Gaussian Fuzzy Sets:

$$A(x; m, \sigma, \mu) = \begin{cases} \exp(-(x-m)^2 / \sigma^2) & \text{if } x \leq m \\ \exp(-(x-m)^2 / \mu^2) & \text{if } x > m \end{cases} \quad (10)$$

Parabolic Fuzzy Sets:

$$A(x; m, p) = \begin{cases} 1 - p^2(x-m)^2 & \text{if } x \in [m-1/p, m+1/p] \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

2.12. Type-2 Fuzzy Sets and Intervals

A fuzzy set of type-n, $n=2, 3, \dots, \infty$, was first proposed by Zadeh [41]. The membership function of such a function ranges over fuzzy set type-1 where the membership function of fuzzy type-1 ranges over $[0, 1]$. Based on this, a fuzzy set type-2, “ \tilde{A} ”, is characterized by a membership function: $\mu_{\tilde{A}} : U \rightarrow [0,1]^J$ where the value of $\mu_{\tilde{A}}(u)$ is called a fuzzy grade and is a fuzzy set which ranges over $[0, 1]$ or in the subset of J of $[0, 1]$ [42].

$$\begin{aligned} \tilde{A} &= \sum_{u \in U} \mu_{\tilde{A}}(u) / u = \sum_{u \in U} \left[\sum_{\mu_i^{(u)} \in J_u} \frac{S_i^{(u)}}{\mu_i^{(u)}} \right] / u, \\ J_u &\subseteq [0,1], \quad 0 \leq S_i^{(u)}, \quad i \in I^{J_u} \end{aligned} \quad (12)$$

Here, I^{J_u} is the index set of J_u that is completely consistent with the reality that $\mu_{\tilde{A}}$ is a fuzzy membership grade. Therefore, $\mu_i^{(u)}$ is the i^{th} value of u with the strength of $S_i^{(u)}$.

$\sum_{\mu_i^{(u)} \in J_u} \mu_i^{(u)}$ forms the main membership and $\sum_{\mu_i^{(u)} \in J_u} \frac{S_i^{(u)}}{\mu_i^{(u)}}$ indicates the fuzzy grade or secondary

membership function of the member u . The amount of change of the secondary membership function can also be called secondary grade.

Therefore, $\left[\sum_{\mu_i^{(u)} \in J_u} \frac{S_i^{(u)}}{\mu_i^{(u)}} \right]$ for any given $u \in U$ is a special type-1 fuzzy set that defines the membership value of u in \tilde{A} .

Domain of Uncertainty (DOU) is a set of $\left\{ \sum_{u \in U} \sum_{i \in J_u} \mu_i^{(u)} / u, S_i^{(u)} > 0 \right\}$ [43, 44]. In the type-2

fuzzy set, DOU does not say much about the strength of each membership degree and only identifies the region of primary membership degrees. However, where the fuzzy set is continuous with naturally ordered primary values, the domain is called the Footprint of Uncertainty or FOU. Thus, we shall use the FOU and DOU interchangeably.

We have observed that a type-1 fuzzy set is a special instance of type-2 fuzzy sets, which for all $u \in U$ the set of primary membership degrees, for instance $\sum_{i \in J_u} \mu_i^{(u)}$ is a singleton with the strength of unity.

Now that we have discussed about type-2 fuzzy sets, we shall introduce the special case of a type-2 fuzzy set which is an interval type-2 fuzzy set.

If $S_i^{(u)} = 1, \forall u \in U$ and $\forall i \in I^{J_u}$ then the type-2 fuzzy set gets down-sized to an interval type-2 fuzzy set.

A discrete interval type-2 fuzzy set is shown by:

$$\tilde{A} = \sum_{u \in U} \left[\sum_{\mu_i^u \in J_u} \frac{1}{\mu_i^{(u)}} \right] / u, \quad J_u \subseteq [0,1], \quad i \in I^{J_u} \quad (13)$$

Therefore, the footprint of uncertainty is defined as:

$$FOU(\tilde{A}) = \bigcup_{u \in U} J_u \quad (14)$$

2.13. Centroid of a Fuzzy Set

Upper membership function (UMF) and lower membership function (LMF) of \tilde{A} where \tilde{A} is a type-1 fuzzy set are also fuzzy sets and represent the upper limit and lower limit of $FOU(\tilde{A})$ respectively. They are defined as:

$$UMF(\tilde{A}) = \overline{\mu}_{\tilde{A}} = \sum_{u \in U} \mu_{n_i}^{(u)} / u, \quad n_i = |I^{J_u}| \quad (15)$$

$$LMF(\tilde{A}) = \underline{\mu}_{\tilde{A}} = \sum_{u \in U} \underline{\mu}_i^{(u)} / u \quad (16)$$

Therefore, we can formulate:

$$J_u = [\underline{\mu}_{\tilde{A}}(u), \overline{\mu}_{\tilde{A}}(u)] \quad (17)$$

$$FOU(\tilde{A}) = \bigcup_{u \in U} [\underline{\mu}_{\tilde{A}}(u), \overline{\mu}_{\tilde{A}}(u)] \quad (18)$$

It is important and notable that FOU has the prime role in characterizing an interval type-2 fuzzy set. The boundary of the endpoints of a centroid which serves as the measure of the uncertainty of an interval type-2 set is defined as follows:

$$c_{LMF} = \frac{\sum_u u \underline{\mu}(u)}{\sum_u \underline{\mu}(u)} \quad (19)$$

$$c_{UMF} = \frac{\sum_u u \overline{\mu}(u)}{\sum_u \overline{\mu}(u)} \quad (20)$$

$$\bar{c}_l = \min(c_{LMF}, c_{UMF}) \quad (21)$$

$$\underline{c}_r = \max(c_{LMF}, c_{UMF}) \quad (22)$$

$$\underline{c}_l = \bar{c}_l - \frac{\sum_u (\bar{\mu}(u) - \underline{\mu}(u))}{\sum_u (\bar{\mu}(u)) \times \sum_u \underline{\mu}(u)} \times \frac{\sum_u \underline{\mu}((u - \text{Inf}(u)) \times \sum_u (\bar{\mu}(\text{Sup}(u) - u))}{\sum_u \underline{\mu}((u - \text{Inf}(u)) + \sum_u (\bar{\mu}(\text{Sup}(u) - u))} \quad (23)$$

$$\bar{c}_r = \underline{c}_r + \frac{\sum_u (\bar{\mu}(u) - \underline{\mu}(u))}{\sum_u (\bar{\mu}(u)) \times \sum_u \underline{\mu}(u)} \times \frac{\sum_u \bar{\mu}((u - \text{Inf}(u)) \times \sum_u (\underline{\mu}(\text{Sup}(u) - u))}{\sum_u \bar{\mu}((u - \text{Inf}(u)) + \sum_u (\underline{\mu}(\text{Sup}(u) - u))} \quad (24)$$

3. Chapter 3: Methodology and Experimental Procedure

3.1. System Design Scheme

Figure 2 shows a schematic of our design. We will describe each stage of the design to show how it works.

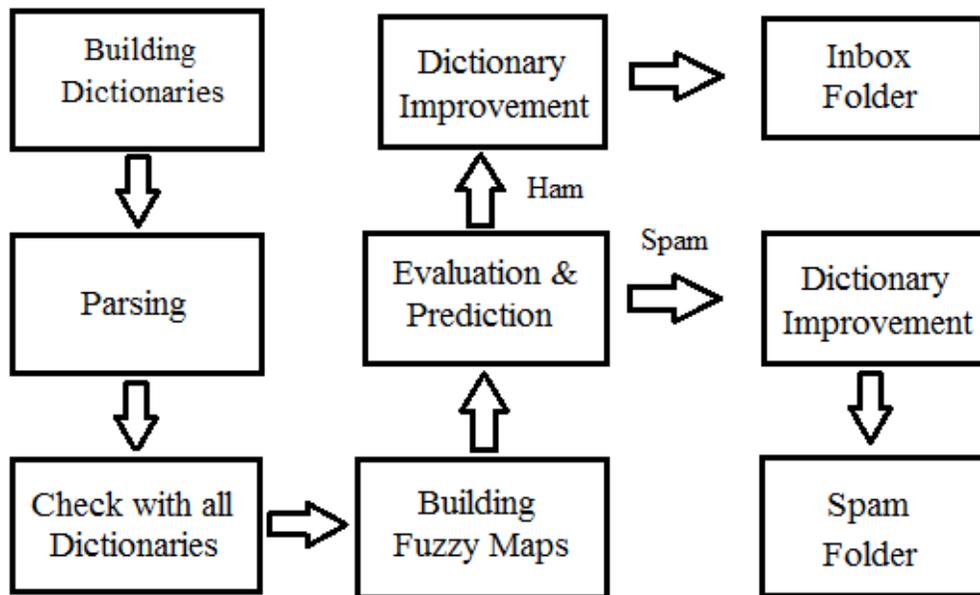


Figure 2. Overall System Architecture

3.2. Building the Dictionaries

A pre-processing procedure needs to be performed at the beginning to build a dictionary for each category of emails. Our design includes one main dictionary for each category of spam and one general white dictionary for all categories. Since we have ten different categories of spam, we need a word dictionary for each one of them. In total, we will use eleven dictionaries, these ten category dictionaries plus one white dictionary.

The first step in the process is to create a white dictionary, or white list. This is the first step to be done. The purpose of this white list is to eliminate some common words that are used in both ham and spam, but that are not necessarily a sign of a spam. The list includes words that are often found in emails but that are not considered to be indications of a spam, such as simple verbs - do, does, etc. - or some common words, such as you, me, etc. We have identified one hundred words that are to be considered white words. In other words, our white dictionary has one hundred elements. Though by no means exhaustive, this list will suffice for our purposes. This white dictionary is built manually.

For each category of spam, we need a word dictionary of common words that are usually used in that particular category. For instance, the word “Sex” will be kept in the Adult category dictionary and the word “Viagra” will be kept in the Health category dictionary. For building our dictionaries, we use fifty spam emails per category. We then calculate the intersection of each single file out of the fifty in that specific category with the union of the rest of the forty nine files. Finally, we compute the union of these intersections and make our dictionaries based upon this final union.

$$\bigcup_{i=1}^n (A_i \cap (\bigcup_{\substack{j=1 \\ j \neq i}}^n A_j)) \quad (25)$$

For $A_i, i = 1, 2, 3, \dots, n$

where A represents the spam files which we have used as our base-line to make the base dictionaries.

We also tried the simple union of all fifty files in the Adult category and we came up with 319 words other than the white. However, when we used our formula we came up with 246 words. After carefully reviewing the acquired words, we learned that the extra words we acquired through simple union were not a sign of a spam in that particular category. Based on this learning, we decided to use this formula for all of the remaining categories.

It needs to be mentioned that we built our dictionaries white word free. This means that after we acquired the words from the file and before using the above formula, we checked the words

against our white dictionary. The white words were eliminated from the list. Then, after using the formula, we constructed the category dictionary with the remaining words.

In our system, each dictionary is a table. Each element of that table is a structure containing five elements: one element for the actual word, one element for the maximum number of which this specific word has been found in any of last fifty files, one element is the count of all such specific word in all files. This element will be used to get the average, or mean, of the frequency of that word. The next element is a list of two numerical elements which holds the upper and lower limits of weight, when the weights get calculated [see section 3.6.2]. The last element is a list of the fifty elements and each element holds the count of the word in each of the last fifty files.

3.2.1. Weight Calculation

For each word in our dictionary, we have a related weight. To calculate this weight, we have a buffer of fifty values for each word, which each value is the count number of that specific word in each of the last fifty files we have worked on recently [see section 3.2].

For example, the word “free” in our dictionary has a buffer containing fifty values. Value number one is the number of instances of the word “free” that have been found in the first file processed. Value number two is the number of instances of the word “free” that we have found in the second file and so on [see section 3.2]. After we gathered the words in our dictionaries, we calculate the weight of each word based on the average, or mean, of their frequency and the underlying standard deviation [52]. We denote the weight as an interval.

$$\mathbf{Weight} = [AVG - STDEV, AVG + STDEV] \quad (30)$$

As we go forward with new emails, we assign the oldest value for each word in our buffer and add the new value to the buffer. However, we have a policy in our system that once a word enters our dictionaries, its weight never goes down to zero. If the last fifty emails we have

processed had zero of that specific word, our system automatically sets the weight of that word to minimum (0.0001). This policy permits adaptation to changes in spam trends. If spammers stop using a specific word for a period of time, our system sets the minimum weight for that word. However, that word is always considered in our fuzzy classifier and when spammers start using that word again, our system raises the weight based on the average and the standard deviation of our buffer.

We could have adopted a larger buffer, i.e. more than 50 values, for our system. However, there is a tradeoff since the larger the buffer becomes, the slower the system also becomes. A small buffer of fifty therefore seems reasonable. We arrived at the number fifty through the process of trial and error.

Standard deviation [52] is calculated as:

If X is a variable with the mean value of μ then:

$$E[X] = \mu \quad (31)$$

Here in the formula, operator E represents the average value of X . Therefore, we have the standard deviation of X as:

$$\sigma = \sqrt{E[(X - \mu)^2]} \quad (32)$$

When X gets random numbers from a limited data set with the same probability, σ (sigma) or the standard deviation is defined as:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2} \quad (33)$$

It should be mentioned that since we calculate the weight based on the mean and standard deviation, there are some cases where the data is quite scattered and the standard deviation value is quite large. In these cases, we arrive at a negative lower limit number for the weight. In order to eliminate this issue from our system we always shift the average by 10 points. In this way, our baseline will rise by 10 point and we will not have negative values for either of our limits. This change does not affect our result since all of the weights are increased by 10 points. This measure merely prevents us from working with negative weights.

3.3. Parsing

The next step is to work on the body of the email. For this goal, we make a linked-list of our text where each node on our list contains two elements: a string element that holds the actual, single word and a numerical element that holds the count of that specific word in the entire email. We also have the ability to move both forward and backward on the list. The reason that a linked-list has been used in our implementation is that working with a linked-list is easy and sufficient for our purposes.

When we produce the linked-list for the first time, the count index for each word is one. Therefore we will have a lot of redundant words in the list. For example we might have forty instances of the word “you” in an email. We will eliminate the redundant words and increase the count index of elements of the word later. We also eliminate all the white words, but keep track of their number. For example, we record how many instances of the words “you” or “me” there are that we eliminated from the linked-list.

Since spammers use different techniques to deceive the spam filters, we need to be careful in the parsing process. One of the common techniques among spammers is to insert spaces between the word letters. In order to neutralize this spamming technique, whenever we see a one letter word we check the next word immediately. If the second word was a one letter word as well, then we check the next word until we find a word without a single letter. Then we combine the whole single letter words together and we see them as one word. It should be mentioned here that the

two single-letter words, “I” and “a”, are already in our white dictionary, however, neither “I” nor “A” will be followed with another single letter in an email. Therefore, these two words would create a problem. Once this first step is complete, we use a function known as a strip function. This function will take each word and remove all of the endings, such as “ing” or “ed”, from the word’s stem. This function also checks for any unusual signs in the word. Thus “L\$o\$V\$V\$e” will be changed to “Love”. A list of unusual signs is provided in appendix 1.

3.4. Checking with all Dictionaries

In this stage, we need to compare the message with the dictionaries. Before doing so, we need to eliminate all redundant words in order to have a sample of each existing word as well as a number that shows how many instances of that specific word have been found in the email. To do this, we keep the first word and remove all other redundant words from the list and increase one point to the count element of that first word. Then, we check all remaining words against the white dictionary and exclude, or flag, all white words from the list and thereby from the search. Now we have a clean list of words, along with their count, which is ready to be checked against the category dictionaries.

We will find out if there are any similarities between the words in the message and the words in the category dictionaries. In our system, any authorized user can activate or deactivate any of the category dictionaries according to his or her preferences. Thus, we only check the words with all of the active dictionaries. Once that has been carried out, we will come to an overall decision based on all checked dictionaries.

For checking similarities, we use the Jaro-Winkler Distance technique [48, 49]. Before deciding to use the Jaro-Winkler distance technique, we tried the Levenshtein Distance [50], the Smith-Waterman Distance [51] and the Jaro-Winkler Distance techniques. Based on the accuracy of the results we observed using each technique, the Jaro-Winkler Distance technique proved to be the most effective of the three. Thus we determined it to be best suited to our purposes.

We define the Jaro distance “ d_j ” of two strings “ s_1 ” and “ s_2 ”, where $|s|$ denotes the length of string “ s ” as following:

$$d_j = \frac{1}{3} \left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m} \right) \quad (26)$$

where:

- “m”: matching characters’ number
- “t”: transpositions’ number

If two characters “s₁” and “s₂” are not farther than the following then we consider them matching:

$$\left\lfloor \frac{\max(|s_1|, |s_2|)}{2} \right\rfloor - 1 \quad (27)$$

The number of transpositions is the number of matching characters of two strings divided by two. It is defined as:

$$t = \frac{m}{2} \quad (28)$$

Jaro-Winkler distance “d_w” for any two strings “s₁” and “s₂” is defined as:

$$d_w = d_j + (l \times p (1 - d_j)) \quad (29)$$

where:

- “dj”: strings s₁ and s₂. Jaro Distance
- “l”: length of common prefix at the start of the string, to a maximum of 4 characters.
- “p”: constant scaling factor for how much the score is adjusted. “p” should not be more than 0.25, or the distance can become larger than 1. The standard value for “p” in Winkler’s paper is p = 0.1 [49].

Using the Jaro-Winkler technique in all comparisons, we first check our linked-list against our dictionaries to find any possible similar patterns. These we call “hit” words if the similarity is 100% and the distance between them is zero. Furthermore, for all words that have a similarity of more than 80% and a distance of less than 0.2 out of 1, we refer to them as “similar” words. If the similarity is less than 80%, we do not name them. In these cases, we simply record the distance. We keep track of all similar, hit and white words along with their associated weights, which are determined by our dictionaries, and their associated distances.

3.5. Building Fuzzy Maps

Fuzzy maps are built based on information that we have already extracted from the email. We use an interval type-2 fuzzy paradigm to decide whether the email is ham or spam.

We build our fuzzy maps based on the distance of each word in the email with our dictionaries, the weight of the closest entries in the dictionary with the word [see section 3.2.1] and the frequency of the use of each word in the email (TF-IDF).

We use a three dimensional (10,000 x 10,000 x 1) matrix with which to build our map. The first and second dimension of the map, or matrix, holds the distance (10,000) and weight values (10,000) and the third dimension of the map, or matrix, holds the TF-IDF value (only 1). It is notable that since our matrix has only 10,000 values for each distance and weight vectors in the map, all values which exceed the limitations of 4 digits in the map must be rounded to the closest value near it. For example if we have a distance (0.345491), it will be rounded to (0.3455). The same is also true for the weights.

First, we calculate or extract each Distance (section 3.5.1), Weight (section 3.2.1) and Weight of each interval (section 3.5.2). Second, we put these each weight intervals on the map on their proper distance values. Third, we associate the third dimension of the map, or all points that are under the actual weight interval, with the weight of that specific interval. Figure 3 shows a sample of the map.

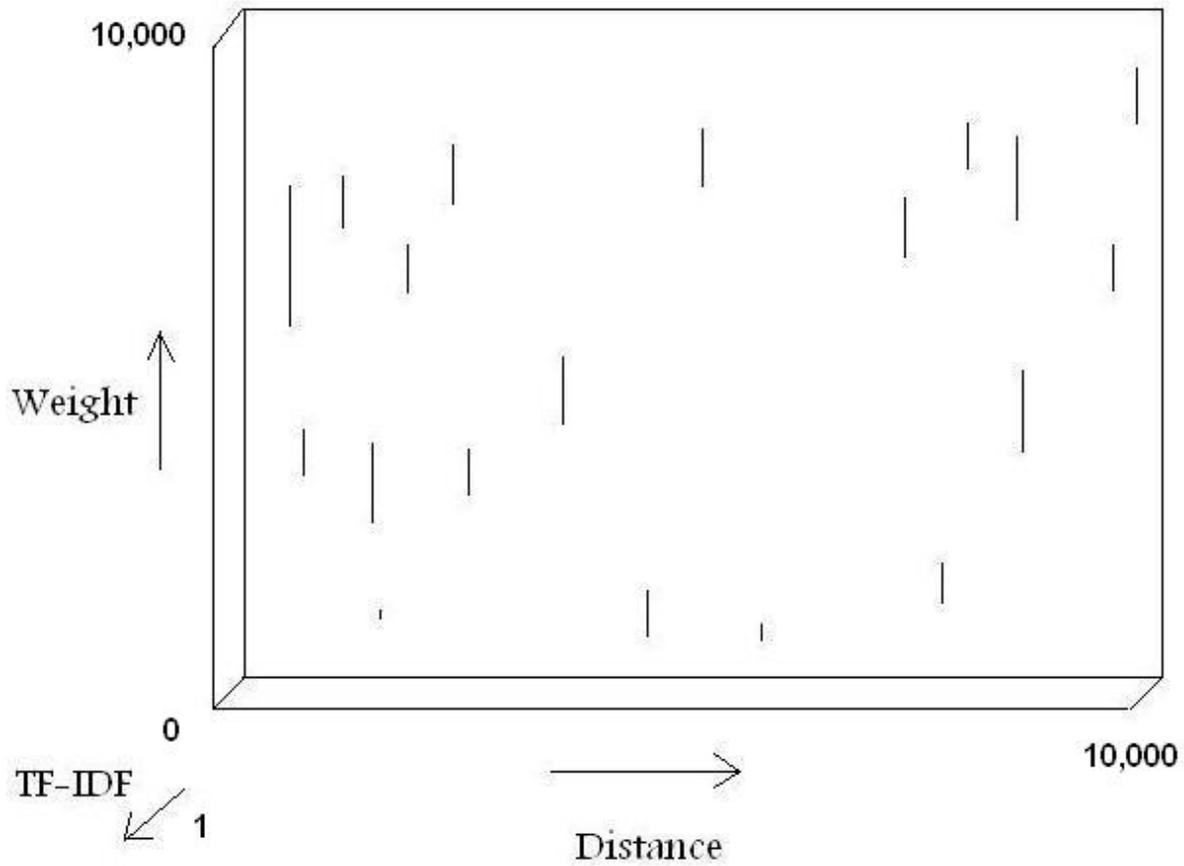


Figure 3. A Sample of the 3D Fuzzy Map

3.5.1. Distance Calculation

To calculate the distance of a word with a dictionary, first we calculate the distance between the word and all dictionary entries using the Jaro-Winkler Technique [48, 49]. If, in any case, we find more than one entry in the dictionary having the same distance with the searched word, we will choose the maximum, or union, weight of those matched entries. We set the maximum distance (1) for all white words.

3.5.2. Weight of Each Interval in the Map (Frequency of each Word)

After building the first and second dimension of the map with distance and weight of each word, it is time for the third dimension, which is the frequency of use of each word in the email. We need to have a specific weight for each word's related interval in the map that represents the frequency of that particular word in the current email. This weight is different from the weight of the word in our dictionary. We use this specific weight as a statistical measure to evaluate and judge the importance of each word to the whole email. This weight or importance of the word increases proportionally to the number of times that the word is used in the whole email. Here, we employ TF-IDF (term frequency-inverse document frequency) technique [53], to calculate the specific weight for each interval in the 3rd dimension of the map.

We define the term frequency ($tf_{i,j}$) as [53]:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (34)$$

Here, the formula $n_{i,j}$ denotes the number of occurrences of the word (t_i) in the text, or document, d_j , and the denominator is the sum of number of occurrences of all words in the document d_j .

By dividing the total numbers of documents by the number of documents that actually contain the word and then obtaining the logarithm of the quotient, we can arrive at the inverse document frequency. This is essentially a measure to show how important a specific word is to the whole document. It is defined as [53]:

$$idf_i = \log \frac{|D|}{1 + |\{d : t_i \in d\}|} \quad (35)$$

With:

- $|D|$ is the total number of texts (document) in the corpus
- $|\{d : t_i \in d\}|$ is the number of documents where the word t_i shows up (which is $n_{i,j} \neq 0$). If the word is not in the corpus, it would lead to an unfortunate division-by-zero. Therefore, this is common to use $1 + |\{d : t_i \in d\}|$

And therefore, we have:

$$(tf - idf)_{i,j} = tf_{i,j} \times idf_i \quad (36)$$

Since the actual weights of each word in the dictionary are an interval itself, it also appears as an interval in our map. Therefore, after calculating the TF-IDF for each word, we need to associate all members of that interval with that specific TF-IDF.

For example, let's say that we have the word "lovers" with the distance (0.1314) and the weight [0.2118, 0.2482] in the map. Since the weight of the word is an interval itself, it covers 364 points of the map as an interval. So the TF-IDF or weight of the interval should be applied to all 364 points that this interval in the map covers. Therefore in the 3rd dimension of the map we fill all 364 points under that interval with the same TF-IDF.

If we have a word in the map with an identical weight and distance in the map, e.g. Distance= 0.1314 and Weight= [0.1300, 0.1500], then we disregard the lesser TF-IDF and put the larger one in the third dimension.

After applying the TF-IDF weights to each related interval in the map, we are ready to calculate the centroid of the map.

3.6. Evaluation and Prediction

To decide if an email is spam or not, after building its interval type-2 fuzzy map, we will calculate its centroid [see section 2.12.]. Here for calculating the centroid of the map, we consider the map as a two dimensional map.

The reason for this is that the centroid formula does not support three dimensional maps. We begin by calculating the centroid of our 2D map. Then we take the third dimension into the account. We will explain this in greater detail further on in this section.

After calculating the centroid, the centroid's formula gives us four values [please see 2.12 section] that are left and right uncertain boundaries. Now for simplicity, we consider the centroid as a horizontal interval in our map which has no clear end point. Instead, each pair of the above mentioned values are intervals which show the end points of that larger interval.

We divide the centroid's domain, which is essentially our distance vector, into three zones. The first zone represents the spam area, and the third zone represents ham area. The second zone is our uncertain area. Figure 3 shows a schematic of the fuzzy subsets. We have come up with these fuzzy subsets through trial and error.

Based on our experiments, the words that were never seen before usually have distances of more than 0.6. Most of the words that were seen before will have a distance that is less than 0.4. Words with a distance between these two values are rare, regardless of whether or not they are spam words. The distribution of the intervals in our maps for several different emails led us to come up with these three zones.

While in our experiments we did not experience any instances of an email ending up in the uncertain zone, we decided to keep it since we might face such email in the future. Since we have had a limited number of spam emails to test, we thought that it is the best to have uncertain zone.

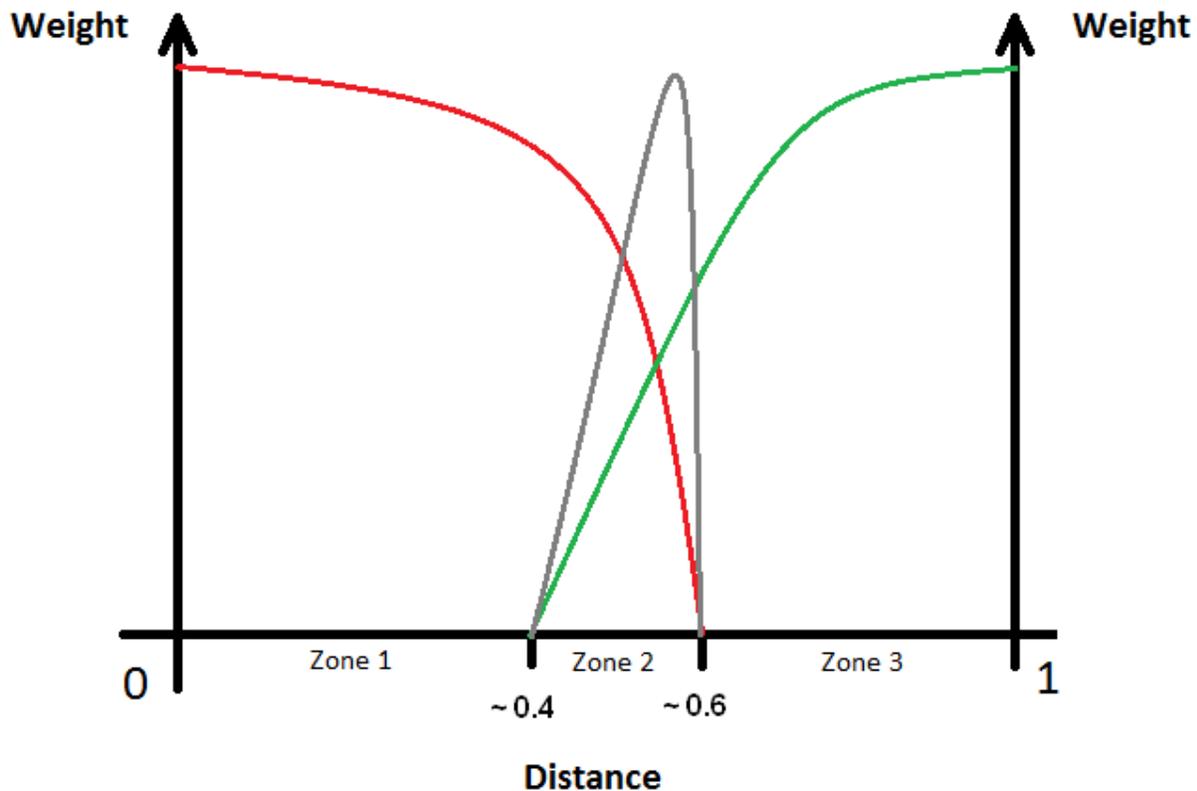


Figure 4. A schematic of the Membership Function of the System

If we have an email whose underlying centroid ends up in second zone, our system wouldn't categorize the email and leave the decision to the user. If user decides that the email is a spam then our system will improve itself by adding the new words of that email and also changing the weights of existing words [see section 3.2.1].

However, in most cases, we have centroids that are big and they cover more than one zone.; Specifically, since the end points of the centroid are intervals with uncertain boundaries, we have had cases where even the end points are in two different zones. It is very difficult to decide which centroid belongs to which zone. In addition, the data elements associated with the centroid in the third dimension of the map can belong to more than one zone. We need to use a sophisticated technique to separate these associated data sets so we can decide which zone they

belong to. In cases where the centroid ends up in a single zone we can decide more easily. Challenges arise when we have to decide between two zones.

Those third dimension values are our guide, or criteria, for determining which zone the centroid belongs in. Furthermore, the number of the points, or the length, that centroid has in each zone is also our guide. To do so, we employ a fuzzy c-means (FCM) clustering technique [54, 55].

In fuzzy clustering, the data sets could belong to different clusters where each specific element is a set of membership levels. This fact indicates the strength of the association between the data sets and a specific cluster. The process of assigning each one of the data elements to one or more clusters by considering these membership levels is called fuzzy clustering.

First we define three clusters; each one belongs to one zone (j). Second, we give each point (x) the degree of belonging to each cluster (u). Third, we repeat the algorithm until it covers all points of the centroid (i). Fourth, using the below formula we calculate the centre of each cluster (c). The next step will be deciding each point belongs to which. The task is done by minimizing the following function:

$$J_m = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m \|x_i - c_j\|^2, \quad 1 \leq m \leq \infty \quad (37)$$

- “ m ”: any real number greater than 1
- “ u_{ij} ”: the degree of membership of “ x_i ” in the cluster “ j ”
- “ x_i ”: the “ i th” of an n-dimensional measured data
- “ c_j ”: n-dimension center of the cluster
- “ $\|*\|$ ”: similarity between any measured data and the centre.

We do the fuzzy partitioning by an iterative optimization of our mentioned objective function with the update of membership “ u_{ij} ” and the cluster centers “ c_j ” by:

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}} \quad (38)$$

$$c_j = \frac{\sum_{i=1}^N u_{ij}^m \times x_i}{\sum_{i=1}^N u_{ij}^m} \quad (39)$$

The mentioned iteration would be stopped when $\max_{ij} \{ |u_{ij}^{k+1} - u_{ij}^k| \} < \varepsilon$, where “ ε ” is a termination criterion between 0 and 1, whereas “ k ” are the iteration steps.

This procedure converges to a local minimum or a saddle point of “ J_m ” [54, 55].

After using the FCM technique, we can finally determine the correct zone for each centroid based on those clusters.

3.7. Dictionary Improvement

Spammers always come up with methods for breaking through spam-filtering techniques. An effective Anti-spam system needs to be adaptive in order to combat new spamming techniques.

Our system has an adaptive nature. It has the capacity to adapt to new words that it has never seen before. If spam trend change over time and spammers try to employ new methods of using words, our system can keep up with these changes by improving its dictionaries regularly and consistently.

We have two different approaches to dictionary improvement:

- Email is predicted to be a ham: In this case, the weight of the hit words will get reduced automatically by applying their weight in the buffer in a negative way. When we put the number of instances of the hit word with a negative in the buffer, it affects the average, or mean and so the standard deviation [see section 3.6.2].

- Email is predicted to be a spam: In this case, the weight of the hit words will get raised automatically [see section 3.6.2]. In addition, after excluding all white and the hit words that we have identified from the email, we will add all the remaining words of the email to our dictionaries.

The weight of the new added words will be very low at first, due to the related average and standard deviation. However, their weight will increase over time if we have more hits for them in the future. After dictionary improvement, our system's objectives are nearly fulfilled. The system then puts the email in the inbox folder if the email is ham or puts it in spam folder if it is spam.

3.8. Results and Analysis

Our system uses the same method to detect spam in all ten categories of spam. Here, we shall perform an experiment to demonstrate that our system works as intended. In our experimental evaluation, we have used the total number of 1895 emails including 567 emails containing spam contents and 1328 legitimate emails. Those 567 emails belong almost to all types of spam but mostly the Adult type.

We also have employed the confusion matrix, or contingency table, to evaluate our system [56]. In a contingency table, true positive (TP) denotes the correct classification of a spam where the false positive (FP) denotes incorrect classification of spam. True negative (TN) is the correct classification for ham and false negative (FN) denotes the incorrect classification of ham. Table 2 shows our contingency table.

		Predicted by our System	
		Positive	Negative
Uncertain Emails = 0			
Real Emails = 1895	Positive	TP = 441	FN = 126
	Negative	FP = 82	TN = 1246

Table 1. Contingency Table

Accuracy (A_{cc}) is the most important evaluator of any spam detection system. This measure evaluates the proportion of correctly classified instances whether to be ham or spam. It is also the general factor of effectiveness of any spam detector system [56]:

$$A_{cc} = \frac{TP + TN}{TP + TN + FP + FN} \quad (40)$$

There are also two other measures which also are equally important in measuring the effectiveness of a spam detection system: Spam Precision (S_p) which denotes the percentage of truly classified messages categorized as spam and Spam Recall (S_r) which denotes the proportion of accurate categorization of real spam messages by the system [56]:

$$S_p = \frac{TP}{TP + FP} \quad (41)$$

$$S_r = \frac{TP}{TP + FN} \quad (42)$$

Table 3 shows the results of our system. Since researchers are more concerned about hams being incorrectly blocked than they are about allowing a spam email to pass through, spam precision is the most important factor in spam filtering. As we can see here, our system has a very good potential to respond to this concern [comparing to 56,57,58,59 and 60].

Total Emails	Actual Spam	Actual Ham	Spam Accuracy	Spam Precision	Spam Recall
1895	567	1328	89 %	84.3 %	77.7 %

Table 2. Main System Results

As we know, there is no such thing called “a spam benchmark” for the purpose of testing. Spammers are trying to change the trend of their spam day by day and there are almost no two spam emails that are 100% similar, at least not in the case of plain text spam, which is what our research is concerned with. Therefore, the results that people present for their work in this field may vary dramatically if they use a different data set. The results we have here are merely based on the testing data set that we could provide.

Since we have had no access to the source code of other text-based spam detection systems with which to test our data set, we decided to test it against two major mail servers: Yahoo Mail Server and Microsoft Mail server, or Hotmail. However, we do know that this test is not one

hundred percent accurate. These mentioned mail servers use different spam detection methods on many different levels. However, we tried to simplify the test as much as possible so it is as close as possible to the plain text level. We did not choose to use Gmail, the Google Mail Server, because we acquired all our spam emails from Gmail. Since Gmail had already detected these emails as spam, there was no reason to use it again.

Based on our past experience, after catching some spam from a single sending account, Yahoo blocks the sending account and starts to send most of the emails from that particular account to the spam folder. That includes the legitimate emails. This shows that Yahoo identifies the sending account and simply blocks it. This also occurs with Hotmail. The only difference between these two servers is the amount of spam that these two consider critical in order to block the sending account. We do not know what that critical number is or whether other factors are involved in the processes.

For the receiving account only one is enough. We used a legitimate receiving account on each of those mail servers which have been in use for a long time.

In order to be as accurate as possible for the sending accounts, we needed to divide our data set into smaller fractions. Each new data set, or fraction, contained a smaller portion of spam and a larger portion of ham. Also, for the purpose of accuracy, we attempted to make each new data set be as similar to the other data sets as possible, in terms of the number of ham versus spam and the category of spam.

In the end, we had twenty new data sets in total. Then, we built and used twenty different sending accounts on each server and then sent those data sets from these sending accounts to that one receiving account on each mail server. The results are shown below.

		Predicted by Hotmail	
		Positive	Negative
Real Emails = 1895	Positive	TP = 389	FN = 178
	Negative	FP = 213	TN = 1115

Table 3. Hotmail Contingency Table

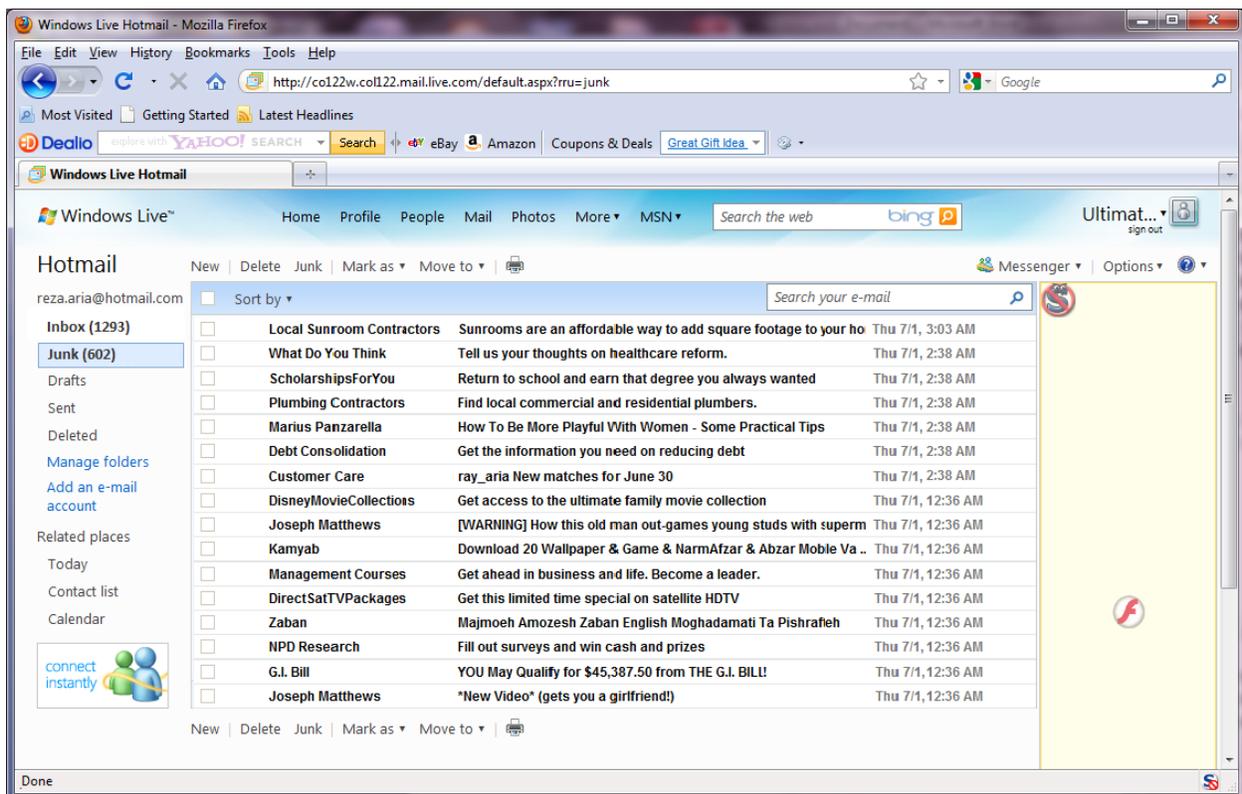


Figure 5. Hotmail Snapshot

Total Emails	Actual Spam	Actual Ham	Spam Accuracy	Spam Precision	Spam Recall
1895	567	1328	79.3 %	64.4 %	68.6 %

Table 4. Hotmail Results

		Predicted by Yahoo	
		Positive	Negative
Real Emails = 1895	Positive	TP = 428	FN = 139
	Negative	FP = 317	TN = 1011

Table 5. Yahoo Contingency Table

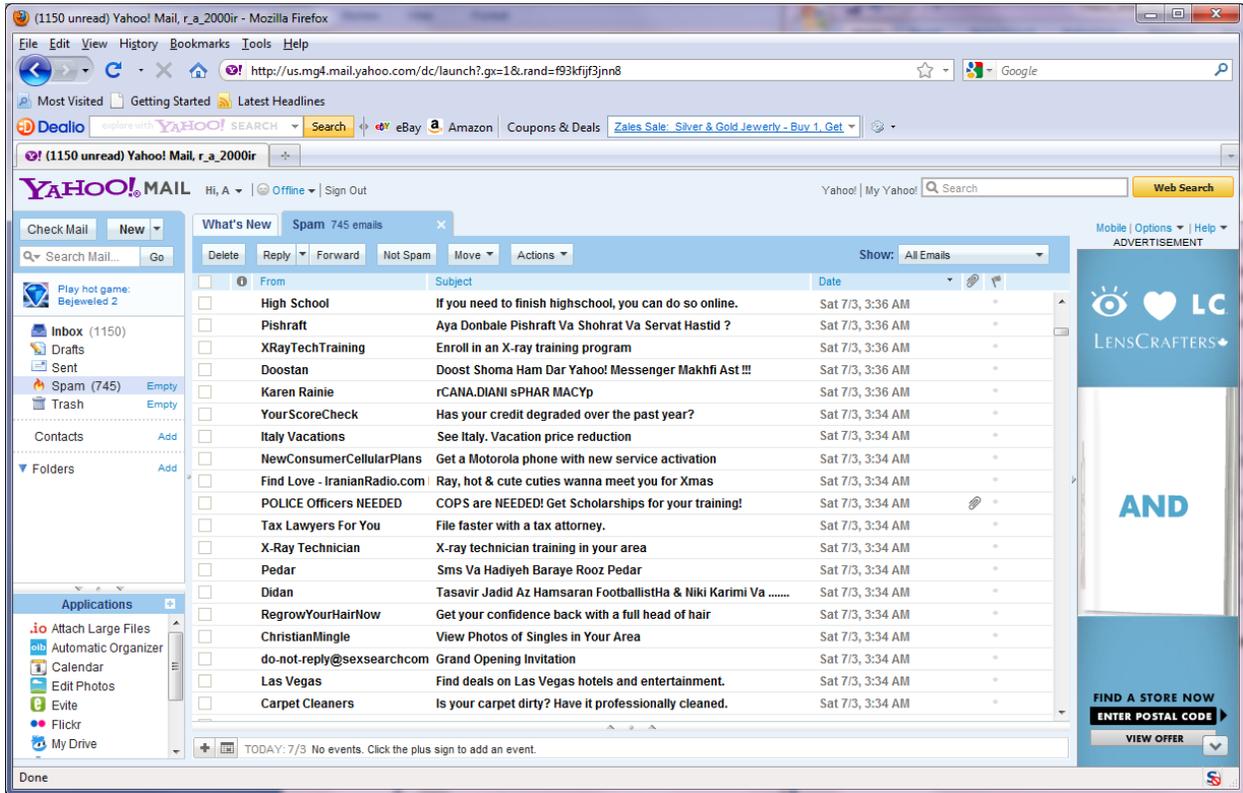


Figure 6. Yahoo Snapshot

Total Emails	Actual Spam	Actual Ham	Spam Accuracy	Spam Precision	Spam Recall
1895	567	1328	75.9 %	57.4 %	75.4 %

Table 6. Yahoo Results

The department of computer science at our university uses spam-assassin (see section 2.8.5) to detect spam. We have also tested our system with spam files that the spam filter of our computer science department (spam-assassin) captured. We have identified and extracted 438 adult spam messages from two sample data sets (spam files) we received from our computer science department. The first data set contained actual spam and the other one contained possible spam each containing 700 messages. Then we fed them to our system to observe how many of them would be identified by our system. Table 7 shows the results.

Total Spam Messages	True Detected Messages	False Detected Messages	Uncertain Messages	Spam Detection Accuracy
438	381	57	0	86.9 %

Table 7. Computer Science Data Set Results

Chapter 4: Conclusion and Future Work

4.1. Conclusion

In this thesis, the goal was to implement a spam-filtering system that is more efficient than currently existing methods. First, we studied all existing methods to understand their strengths and weaknesses. Based on our studies, the existing methods are reliable and efficient. However, each method has weaknesses as well. Some methods require periodic maintenance and updates while others are not as efficient and practical. Our objective was to improve upon the as many of the existing methods' flaws as possible.

We decided to use fuzzy type-2 method in order to be able to handle more uncertainty. The type-1 fuzzy spam detection systems may still have room for improvement, but any improvements would be minor. On the other hand, fuzzy type-2 systems are capable of handling uncertainty quite well. Hence, these systems provide us with more power to deal with a wider variety of issues. It is quite reasonable to move from fuzzy type-1 to fuzzy type-2 systems in order to see major improvements. So it was our motivation to propose a better method, which has the strengths of the existing methods, yet fewer of their weaknesses and which would also be more efficient than existing methods. We have considered the interval type-2 fuzzy sets to tackle this problem. This technique has not been used with its full capacity in this field and we think that it has the capability to satisfy us with our expectations.

Here are a few strength points of our system:

- It gives the authorized users the ability to manually choose not to block a certain type of spam while blocking the other types, simply by excluding the related dictionary to that type of spam in the search.

- It is flexible and adapts to new words and spamming techniques, such as leaving spaces or sign between letters of a word, which spammers may use to confuse anti-spam systems.
- It is able to strip the words to their closest possible roots by removing the endings from their stems. This is functionality will allow it to function with the other languages that have the similar stems to those in the English language.
- It has the ability to learn from spam and spammers, insofar as it incorporates new words from emails that are identified as spam into its dictionaries. Thus, it will be able to identify those new words in the future. If the spammers try to change the techniques used in their spam, the system will self-adapt to the new trends.
- The new invented way of weight calculation provides more flexibility and accuracy.
- It is very capable of dealing with new and unseen spam.

We performed a comprehensive experiment on spam categorization. Our results imply that interval type-2 fuzzy set combined with fuzzy clustering technique is an efficient technique for detecting and categorizing spam compared to the existing fuzzy methods [56, 57, 58, 59 and 60]. Also, our test results from two popular mail servers, Yahoo and Hotmail and the mail server of our computer science department support our position. Therefore, the proposed method is considered to be a reliable and efficient method.

4.2. Future Work

In future work, we would like to extend our research to filter email messages with an HTML body. With an effective spam filtering mechanism for plain text, the next logical step is adding additional components that can parse HTML messages as well. In addition, we will do further research on how to find more features and aspects having high differential power to improve the performance of our system.

The other future prospect for our research is to extend our work to mobile devices and cellular networks. There are millions of people using cellular devices and gadgets. Spammers are trying to use the Short Messaging System, SMS, or the Multi-Media Messaging System, MMS, to spam users. Extending our efficient system to function with mobile devices and cellular networks would also be a logical next step.

Lastly, we are planning to extend our research to move from Interval Type-2 Fuzzy Sets and implement a new spam detection system based on General Type-2 Fuzzy Sets which are more complicated to use but which have the added benefit of using a full three dimensional space instead of limited three dimensional comparing to Interval Type-2 Fuzzy Sets.

Appendix I. The Unusual Signs in the Words

{ '~', '!', '@', '#', '\$', '%', '^', '&', '*', '(', ')', '-', '_', '+', '=',
{', '['', ']', '|', '\\', ':', ';', '"', '<', '>', '.', '?', '/',
'0', '1', '2', '3', '4', '5', '6', '7', '8', '9' }

ASCII codes below 27 if occurs in between of the letters of a word.

All of ASCII codes over than 128.

Appendix II. White Dictionary Words List

A , AS , I , I'M , I'VE , I'LL , ME , MY , YOU , YOU'LL , YOUR , YOU'RE , YOURS , HE , HE'LL , HIM , HIS , SHE , SHE'LL , HER , HERS , WE'LL , OURS , THEY , THEY'LL , THEM , THEIR , THEIRS , NOT , AM , ARE , IS , ISNT , ARENT , WAS , WASN'T , WERE , WERENT , CAN , CANT , COULD , COULDN'T , MAY , MIGHT , SHALL , SHOULD , SHOULDN'T , MUST , MUSTNT , WILL , WOULD , WOULDN'T , WONT , THE , AND , IN , OUT , BE , BEING , WITH , BY , THIS , THESE , THAT , THOSE , IT , IT'S , IT'S , HERE , THERE , TOP , BOTTOM , NORTH , SOUTH , EAST , WEST , HAVE , HAVING , HAVENT , HAD , HADNT , DO , DON'T , DOING , DOES , DOESN'T , DID , DIDN'T , FOR , FROM , OF , TO , THEN , GET , GETTING , EMAIL , WE , NO , OUR , US

Appendix III. Adult Dictionary Word List

HIKI, LIGHT, EYES, LOVE, GIVE, NEW, ON, LIFE, WANT, DOGYSTYL, MYSELF, PROMISE, HOPE, MOST, ALL, BECAUSE, FEEL, SAFE, RAPE, IF, SOMETH, ROMANTIC, BUT, KNOW, ARMS, TOGETHER, EVERYTH, DREAM, SO, FLIRT, MUCH, BETTER, THAN, EACH, OTHER, EVERY, TIME, LOOK, INTO, GROW, DAYS, GO, SWEET, LAST, TURN, FRIEND, GIRLFRIEND, PULL, BACK, TODAY, DETAILS, HOW, BASICALLY, WHEN, ALWAYS, START, FLIRTS, GOOD, MORE, HELP, HAS, ALREADY, FRIENDSHIP, COMES, TWO, FIRST, YOURSELF, SEX, BEEN, LONG, CHEMISTRY, THINK, NOW, HAPPENS, SHOWS, OFF, BODY, UP, SOME, BIT, EXCIT, ABOUT, PROBABLY, HERE'S, ANOTHER, EXAMPLE, SEE, LOT, OVER, PERIOD, NOTICE, PERSON, SAY, MEET, AGAIN, WHAT, LINE, WHILE, OFTEN, HIGHER, CHANCE, RELATIONSHIP, FUTURE, JUST, KEEP, FRIENDS, WOMAN, HERSELF, CHANG, HI, JOIN, SITE, POST, PICS, ONE, CHAT, PLEASE, VISIT, PAGE, CONTACT, TURNS, LIVE, MILES, APART, VIEW, PROFILE, HELLO, DEAR, COME, LET, NIGHT, MAKE, HEART, BEFORE, HAND, LEAVE, BECOUSE, SOMETIMES, FIND, HOLD, READY, TRY, TAKE, TOO, REMEMBER, NEAN, LET'S, GIRL, FLOOR, STAY, SAME, GUYS, DRUNK, THEM, SHARE, ALSO, GIV EVEN, PEOPLE, WAY, SITUATION, THERE'S, NATURAL, RAPPORT, PARTY, SHOW USUALLY, FUN, SCAR, MISS, ALLOWS, RUN, AWAY, HURT, FANTASY, HELPS, FANTASIZE, THINGS, EXPERIENCE, NOTH, UNTIL, MANY, EXPECT, WOMEN, SPEND, PLAY, TELEVISION, HOME, LIKE WORKS, RIGHT, SECOND, KEY, GIRLS, NEVER, MATTER, END, WITHOUT, SPARK, AROUND, DAT, BETWEEN, MAN DEEP, UNDERSTAND, TILL, WAIT, SENT, BECOME, DATE, ASK, ANSWER, USE, LOWER, RESPECT, ATTRACTION, WORRY TEACH, QUESTION, BOTH, HEAR, BEAUTIFUL, COURSE, TELL, GOT, ACTUALLY, BUILD, SMART, NEXT, ANIMAL, INCREASE, FALL, REJECTION, WHY, WORK, WELL, VERY, NEWSLETTER, ARTICLES, SEND, DISCREET, ADULT, MESSAGE, CHANGE, TOPICS, TECHNIQUE, ANY, LATER, REALLY, CONVERSATION, AFTER, WRONG SOON, FREE, YEARS, FINE, CHECK, NTEREST, ONLY, KIND, HONEST, TRUTH, IDEAS, WHO, GENERAL, MUTUAL SINCE, REGULARLY, SPECIAL, SOMEONE, OFFER, SEARCH, PICTURES, TALK, ACTIVITY, SUCH, LEADS UNFORTUNATELY, CURRENTLY, PARTNER, TRI, QUESTIONS, POINT, POTENTIAL, ACT, ALONG, SPAM, SAID, PRESIDENT, GOTTEN, TROUBLE, PAST, MARKET, SITES, FEATUR, BESTIALITY, SUBJECT, LINES, NAK, WHICH, ABUSE, COMPLAINTS, DOGS, PORN, WOMEN'S, MONEY, IMAGERY, FEW, CURIOSITY, WE'RE, CHILD, INVOLV, PORNOGRAPHY, NOWN, FANTASIES, REAL, SEXUAL, ASSAULT, SLIPP, SITE'S, CAUSE, TOOK, CONVERSATIONS INTERESTS, EXPERIENC, PUCY, PLAYFUL

Appendix IV. Other Dictionaries

Since we didn't have many non-adult spam emails we only took them to the account to test the different dictionaries of the filter.

Financial

SAVE, MONEY, BIG, DOLLAR, FREE, GRANT, PAID, PAY, MAIL, EASY, AUHTORIZ, HAPPY, LOAN, INVEST, MORGAG, BILL, DEBT, BUSINESS, ATTORNEY, LEADER, TAX, LAWYER, FOOTAGE, SQUARE, LOCAL, ONLINE, CHECK, CHEQUE, COME, SATISFY, INVESTMENT, BANK

Health

VIAGRA, PHARMACY, ONLINE, GOOD, DEAL, CHEAP, EASY, FAST, DRUG, STORE, BEST, HAPPY, APRIL, CHEEP, NONDRUGS, HEALTHCARE

Internet

CASH, PRIZE, WIN, ONLINE, ZABAN, NARMAFZAR, HIGH, VERY, SCHOOL, TERM, SURVIVE, AX, ABZAR, MOBILE, REDUC, ACCESS, FAMILY, DOWNLOAD, UNLIMIT, BASE, WEB, KETAB, DEGREE, SURVEY, COURSE, AREA, TRAIN, COPS, FAST, AWESOME, EASY, REPLICA, WATCH, HOTEL, ENTERTAINMENT, HADIYEH, PEDAR, ROOZ, PLUMBER, RACH, COPY, TARGET, ESPEND, TAKE, APPEND, TAKE, TOOP, INFORMATION, MANAGEMENT, SADE, AYA, KHOOB, SUNROOMS, SCHOLARSHIP, COMMERCIAL

References

- [1] G. Chapman, “Monty Python’s Flying Circus. Just the Words,” vol. 2, chapter 25, Pages 27–28, Methuen Publishing Ltd, 1999.
- [2] S. Atkins, “Size and cost of the problem,” In Proceedings of the Fifty-sixth Internet Engineering Task Force (IETF) Meeting. SpamCon Foundation, pp. ???-???, March 2003.
- [3] J. Postel, “RFC706: On the Junk Mail Problem,” Technical report, Network Working Group, November 1975.
- [4] J. Klensin, “RFC2821: Simple Mail Transfer Protocol,” Technical report, AT&T Laboratories, April 2001.
- [5] P. Resnick, “RFC2822: Internet Message Format,” Technical report, QUALCOMM Incorporated, April 2001.
- [6] B. McWilliams, “Spam Kings”, O’Reilly Media Inc., 2004.
- [7] J. Posluns, “Inside the Spam Carte”, Syngress Publishing, Inc., 2004.
- [8] S.L. Pfleeger, G. Bloom, “Canning spam: Proposed solutions to unwanted email,” IEEE Security & Privacy, vol. 3. ,no. 2, pp. 40–47, Mar-Apr 2005.
- [9] E. Harris, “The next step in the spam control war: Greylisting,” World Wide Web, <http://projects.puremagic.com/greylisting/whitepaper.html>, 2003.
- [10] M. W. Wong, “Spf overview,” Linux J., 2004(120):2, 2004.

- [11] Yahoo! Inc, “Domainkeys: Proving and protecting email sender identity,” World Wide Web, <http://antispam.yahoo.com/domainkeys>, 2010.
- [12] J. Posluns, “Inside the Spam Cartel”, Syngress Publishing Inc., 2004.
- [13] V. Prakash, “Razor: spam should not be propagated beyond necessity”, World Wide Web, <http://razor.sourceforge.net>, 2010.
- [14] P. Deepak, S. Parameswaran, “Spam filtering using spam mail communities”, In Proceedings. The 2005 Symposium on Applications and the Internet, pages 377–383, 2005.
- [15] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, P. Samarati, “An open digest-based technique for spam detection”, In Proceedings of the 2004 International Workshop on Security in Parallel and Distributed Systems, 2004.
- [16] Rhyolite Software, “Distributed checksum clearinghouse”, World Wide Web, <http://www.rhyolite.com/anti-spam/dcc>, 2010.
- [17] N. Lugaresi, “European union vs. spam: A legal response”, In Proceedings of the First Conference on Email and Anti-Spam, CEAS’2004, 2004.
- [18] P. Graham, “Better bayesian filtering”, Spam Conference at Cambridge, MA, 2003.
- [19] Cipher Trust, “Enterprise email security, Spam: a security issue”, a white paper, http://www.visus-it.com/whitepapers/IronMail/spam_security_issue.pdf
- [20] S. Mane, J. Srivastava, San-Yin Hwang, and J. Vayghan, “Estimation of false negatives in Classification”, In Proceedings Fourth IEEE International Conference on Data Mining, pages 475-478, 2004.

- [21] A. Bargiela, and W. Pedrycz, “Granular Computing: An Introduction,” chapter 2-3, pp. 19-80, 2003.
- [22] G.A. Grimes, “Compliance with CANSPAM act of 2003”, *Communication of the ACM*, 50:55–62, 2007.
- [23] ITU, “ITU survey on anti-spam legislation worldwide” Available at <http://www.itu.int/osg/spu/spam/> , 2005.
- [24] E. Moustakas, C. Ranganathan, and P. Duquenoy. “Combating spam through legislation: A comparative analysis of the U.S. and European approaches”, In *Proceedings of Second Conference on Email and Anti-Spam, CEAS’2005*, 2005.
- [25] T. Fawcett, ”In vivo spam filtering: a challenge problem for data mining”, *KDD Explorations*, 5(2):140–148, 2003.
- [26] J. Chan, I. Koprinska, J. Poon, “Co-training on textual documents with a single natural feature set” In *Proceedings of the Ninth Australasian Document Computing Symposium (ADCS 2004)*, 2004.
- [27] I. Androutsopoulos, E. Magirou, D. Vassilakis, “A game theoretic model of spam e-mailing”, In *Proceedings of Second Conference on Email and Anti-Spam, CEAS’2005*, 2005.
- [28] E. Michelakis, I. Androutsopoulos, G. Paliouras, G. Sakkis, P. Stamatopoulos, “Filtron: A learning-based anti-spam filter”, In *Proceedings of the First Conference on Email and Anti-Spam, CEAS’2004*, 2004.
- [29] W. Yih, J. Goodman, G. Hulten, “Learning at low positive rates”, In *Proceedings of the Third Conference on Email and Anti-Spam, CEAS’2006*, 2006.

- [30] I. Androutsopoulos, J. Koutsias, K.V. Chandrinos, C.D. Spyropoulos, “An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages”, Proceedings of the 23rd Annual International ACM Conference on Research and Development on Information Retrieval, 2000.
- [31] X. Carreras, L. Marquez, “Boosting trees for anti-spam email filtering”, In Proc. of RANLP, 2001.
- [32] SpamAssassin, www.spamassassin.org, 2004.
- [33] M. Sahami, S. Dumais, D. Heckerman, E. Horvitz, “A Bayesian Approach to Filtering Junk E-Mail. In Learning for Text Categorization”, AAAI Workshop, pp. 55-62, Madison Wisconsin, 1998.
- [34] H. Drucker, D. Wu, V. Vapnik, “Support vector machines for Spam categorization.”, IEEE-NN, Vol. 10, No.5, pp. 1048–1054,1999.
- [35] J.G. Hidalgo, M. Spez, E. Sanz, “Combining text and heuristics for cost-sensitive spam filtering” In Proc. of CONL, 2000.
- [36] W. Daelemans, Z. Jakub, K.V. Slood, A.V. Bosch, “TiMBL: Tilburg Memory Based Learner, version 2.0, Reference Guide”, ILK, Computational Linguistics, Tilburg University, 1999.
- [37] J.M. Mendel, “Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions”, Upper-Saddle River, NJ: Prentice-Hall, 2001.
- [38] L.A. Zadeh, “The Concept of a Linguistic Variable and Its Application to Approximate Reasoning–1,” Information Sciences, vol. 8, pp. 199-249, 1975.

- [39] P. Graham, "Better Bayesian Filtering", In Proceedings of Spam Conference <http://spamconference.org/proceedings2003.html>, 2003.
- [40] A. Cournane, and R. Hunt, "An Analysis of the Tools Used For the Generation and Prevention of Spam", Computer and Security, Vol. 23, pp 154-166, 2004.
- [41] L.A. Zadeh, "The Concept of a Linguistic Variable and Its Application to Approximate Reasoning-1," Information Sciences, vol. 8, pp. 199-249, 1975.
- [42] M. Mizamoto, and K. Tanaka, "Some properties of fuzzy sets of type-2", Inform. And Control, 31, pp. 312-340, 1976.
- [43] J. M. Mendel, and R.I.B. John, "Footprint of uncertainty and its importance to type-2 fuzzy sets," in Proc. 6th IASTED Int. Conf. Artificial Intelligence and Soft Computing, Banff, Canada, pp. 587-592, July 2002.
- [44] J.M. Mendel, R.I.B. John, and F. Liu, "Interval Type-2 Fuzzy Logic Systems Made Simple", IEEE Trans. Fuzzy Systems, Vol. 14, No. 6, pp. 808-821, 2006.
- [45] J.M. Mendel, and H. Wu, "Centroid uncertainty bounds for interval type-2 fuzzy sets: forward and inverse problems," Proc. of IEEE Int'l. Conf. on Fuzzy Systems, Budapest, Hungary, July 2004.
- [46] State of Spam and Phishing Monthly Report, Symantec Corp, March 2010
- [47] P. Resnick, "RFC2822: Internet Message Format", Technical report, QUALCOMM Incorporated, April 2001.

- [48] W.E. Winkler, "Overview of Record Linkage and Current Research Directions". Research Report Series, RRS, 2006.
- [49] M.A. Jaro, "Advances in record linkage methodology as applied to the 1985 census of Tampa Florida", Journal of the American Statistical Society 84 (406): 414–20, 1985.
- [50] W.J. Heeringa, "Measuring dialect pronunciation differences using Levenshtein distance" Book, 2004.
- [51] T.F. Smith, and M.S. Waterman, "Identification of Common Molecular Subsequences", Journal of Molecular Biology 147:195–197, 1981.
- [52] Y. Dodge, "The Oxford Dictionary of Statistical Terms", Oxford University Press, ISBN 0-19-920613-9, 2003.
- [53] J. Ramos, "Using TF-IDF to Determine Word Relevance in Document Queries", Department of Computer Science, Rutgers University, (University Technical Report).
- [54] J.C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c -means clustering algorithm", Computer & Geoscience, vol. 10, no. 2-3, pages 191-203, 1984.
- [55] U. Kaymak, and M. Setnes, "Extended Fuzzy Clustering Algorithms", ERIM Report Series Reference No. ERS-2001-51-LIS, 2000.
- [56] M.M. Fuad, D. Deb, and M.S. Hossain, "A trainable fuzzy spam detection system", Proc. of the 7th Int. Conf. on Computer and Information Technology, 2004.
- [57] J.W. Kim, S.J. Kang, and B.M. Kim, "A fuzzy inference method for spam-mail-filtering", Advances in Artificial Intelligence, pp. 5-9, Sydney, Australia, 2005.

- [58] E.M. El-Alfy, and F.S. Al-Qunaieer, "A fuzzy similarity approach for automated spam filtering", Proc. of IEEE International Conf. on Computer Systems and Applications (AICCSA'08), April 2008.
- [59] Y. Hu, C. Guo, X. Zhang, Z. Guo, J. Zhang, and X. He, "An Intelligent Spam Filtering System Based on Fuzzy Clustering", in Proc. Of the Sixth International Conference on Fuzzy Systems and Knowledge Discovery, vol. 07, pp. 515-519, 2009.
- [60] W. Menzhen, L. Zhitang, and Z. Sheng, "A method for spam behavior recognition based on fuzzy decision tree", in Proceedings of the Ninth IEEE International Conference on Computer and Information Technology, China, October 2009.
- [61] H. Tahayori, A. Visconti, and G.D. Antoni, "Augmented Interval Type-2 Fuzzy Sets Methodologies for Email Granulation", 2007 IEEE International Conference on Granular Computing (GRC 2007), pp. 139, 2007.
- [62] A. Wiehes, "Comparing Anti Spam Methods", Master's Thesis in Information Security, Department of Computer Science and Media Technology, Gjøvik University College, Norway, 2005.