UNIVERSITY OF PARDUBICE
FACULTY OF ECONOMICS AND ADMINISTRATION
INSTITUTE OF SYSTEM ENGINEERING AND INFORMATICS

MACHINE LEARNING TECHNIQUES IN SPAM FILTERING

DISSERTATION THESIS

Author: Ing. Aliaksandr Barushka
Supervisor: doc. Ing. Petr Hájek, Ph.D.

Pardubice 2020

Abstract

The rapid growth of unsolicited and unwanted messages has inspired the development of many anti-spam methods. Machine-learning methods such as Naïve Bayes, support vector machines or neural networks have been particularly effective in categorizing spam/non-spam messages. In order to further enhance the performance of review spam detection, I propose a novel content-based approach that considers both bag-of-words and word context. More precisely, the proposed approach utilizes *n*-grams and the Skip-Gram word embedding method to build a vector model. As a result, high-dimensional feature representation is generated. To handle the representation and classify the spam accurately, ensemble learning techniques with regularized deep feed-forward neural networks as base learners are used in order to overcome slow optimization convergence to a poor local minimum and overfitting issues. In order to verify the proposed approach, I use seven different types of datasets from different spam filtering domains. I show that the proposed spam filtering model outperforms existing methods in terms of classification accuracy, false negative and false positive rates, F-score, area under ROC and misclassification cost. The only drawback of the proposed algorithm is its higher computation complexity.

AUTHOR’S DECLARATION

I hereby certify that: This thesis was prepared separately. All the literary sources and the information I used in the thesis are listed in the bibliography.

I got familiar with the fact that the rights and obligations arising from the Act No. 121/2000 Coll., Copyright Act, apply to my thesis, especially with the fact that the University of Pardubice has the right to enter into a license agreement for the use of this thesis as a school work pursuant to § 60, Section 1 of the Copyright Act, and the fact that should this thesis be used by me or should a license be granted for the use to another entity, the University of Pardubice is authorized to claim a reasonable contribution from me to cover the costs incurred during making of the thesis, according to the circumstances up to the actual amount thereof.

I am aware that my thesis will be accessible to the public in the University Library and via the Digital Library of the University of Pardubice in agreement with the article 47b of the Act No. 111/1998 Coll., on Higher Education Institutions, and on the Amendment and Supplement to some other Acts (the Higher Education Act), as subsequently amended, and with the University of Pardubice’s directive no. 7/2019.

In Pardubice 30. 3. 2020 Ing. Aliaksandr Barushka

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor doc. Ing. Petr Hájek, Ph.D for the continuous support of my PhD study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my PhD study.

Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

Acc	accuracy
AIS	artificial immune system
AUC	area under receiver operating characteristic curve
BERT	bidirectional encoder representations from transformers
BoW	bag-of-words
CBOW	continuous bag-of-words
CNN	convolutional neural network
DFFNN	deep feed-forward neural network
DNN	deep neural network
DT	decision tree
ELM	extreme learning machine
FA	firefly algorithm
FDA	factorial design analysis
FN	false negative
FNR	false negative rate
FP	false positive
FRP	false positive rate
FS	feature selection
GAN	generative adversarial network
GRNN	general regression neural network
IG	information gain
IL	incremental learning
k-NN	*k*-nearest neighbor
LCGM	latent class graphical model
LDA	latent dirichlet allocation
LIWC	linguistic inquiry and word count
LR	logistic regression
LSTM	long short-term memory
MC	misclassification cost
MDL	minimum description length

NB	Naïve Bayes
NN	neural network
PCA	principal component analysis
POS	part-of-speech tagging
ReL	rectified linear unit
RF	random forest
ROC	receiver operating characteristic
RSS	random subspace
SAGE	sparse additive generative model
SGD	stochastic gradient descent
SMO	sequential minimal optimization
SMS	short message service
SSL	semi-supervised learning
SVM	support vector machine
SWNN	sentence weighted neural network
tf	term frequency
tf.idf	term frequency–inverse document frequency
TN	true negative
TNR	true negative rate
TP	true positive
TPR	true positive rate
WOA	whale optimization algorithm
W2V	Word2Vector

Introduction

Spam can be defined as an unsolicited and unwanted message sent electronically by a sender that has no current relationship with the recipient (Cormack, 2006). There exist several subsets of electronic spam. Indeed, spam message can be sent over multiple communication channels, such as e-mail, SMS, social networks or shopping online platforms. E-mail spam consumes users’ time, as users must identify and remove undesired messages; it also takes up limited mailbox space and buries important personal e-mails (Zhang et al., 2004). Meanwhile, SMS spam is typically transmitted over a mobile network (Delany et al., 2012). Recently, social network spam has received increased attention from both researchers and practitioners due to both the considerable number of spammers and the potential negative effects of social network spam on convenience and understanding of all the followers (Zhou et al., 2014). Review volume and review valence have been reported to be significant determinants of retail sales in a meta-analysis of more than 20 empirical studies (Floyd et al., 2014). This is particularly relevant for high-involvement products that can only be reviewed upon consumption. Consumers’ experience of product use is therefore an important assumption. As shown in a recent survey (BrightLocal, 2018), more than 80% of consumers trust online reviews as much as they trust personal recommendations. This is why a considerable attention is given to spam filtering in the above communication channels.

Spam messages can be filtered either manually or automatically. Obviously, manual spam filtering by identifying spam message and removing it is a time-consuming task. Moreover, spam messages may contain a security threat, such as links to phishing web sites or servers hosting malware. Therefore, over a number of decades researches and practitioners have worked on improving automatic spam filtering algorithms. Machine learning techniques are particularly known to be highly accurate in detecting spam messages. The main concept of the machine learning algorithms is to build a word list and assign a weight to each word accordingly. However, spammers tend to include common legitimate messages into the spam message in order to decrease the probability of being detected. There is a number of existing machine learning algorithms applied to spam filtering, such as neural networks (NNs) (Barushka and Hajek, 2016), support vector machines (SVMs) (Bhowmick and Hazarika, 2018), Naïve Bayes (NB) (Almeida et al., 2011) and random forest (RF) (Choudhary and Jain, 2017).

According to the survey by Kaur et al. (2018), ensemble learning methods, such as bagging and random forest, outperform traditional single classifiers. The ensemble methods combine the predictions of several base machine learning algorithms in order to improve accuracy and robustness over single algorithms. In previous studies, ensemble methods employed traditional classifiers like decision trees to effectively filter spam messages. However, surprisingly little attention has been paid to NNs with ensemble learning. Recent evidence showed that NNs equipped with regularization techniques may be highly accurate in detecting e-mail and SMS spam (Barushka and Hajek, 2016). This can be attributed to better optimization convergence and resistance to overfitting. To take advantage of these qualities, this dissertation thesis integrates regularized NNs with ensemble learning methods for automatic spam filtering. In order to further enhance the performance of the proposed algorithm, rectified linear units and dropout regularization are used in deep feed-forward NNs (DFFNNs) in order to address the optimization convergence to a poor local minimum challenge which is common for the traditional shallow NN model.

Generally, spam filtering task belongs to binary classification problem, each message should be identified either as spam or ham. In addition to high accuracy, the spam filtering algorithms should also perform well when it comes to false positive ratio (legitimate message is classified as spam) to avoid situations where legitimate message is not delivered to the intended receiver. Moreover, using accuracy, a traditional classification performance measure does not take account of different costs associated with type I and type II errors. Using accuracy for often highly imbalanced spam datasets might also lead to erroneous conclusions because the minority class (usually the class of spam messages) has little effect on accuracy compared to the majority class of legitimate messages. Therefore, multiple performance measures must be considered when evaluating the spam filtering algorithms.

As noted above, the main idea behind content-based machine learning models is to build a word (phrase) list and assign a weight to each word or phrase (bag-of-words) or word category (part-of-speech tagging or psycholinguistic) (Crawford et al., 2015). However, such features suffer from sparsity, which makes it difficult to capture semantic representation of messages. To address this issue, Ren and Ji (2017) proposed a gated recurrent NN model to detect review spam. This approach utilized word embeddings obtained by using the CBOW (continuous bag-of-words) model (Mikolov et al., 2013; Le and Mikolov, 2014) so that words are mapped to vectors based on their context. Thus, global semantic information can be obtained, and, to certain degree, the problem of scarce data is overcome. This approach was reportedly more effective than traditional bag-of-words or part-of-speech tagging (Lilleberg et al., 2015). Inspired by these recent findings, this dissertation thesis utilizes word embeddings to obtain the semantic representation of e-mails, SMS, social network messages and online reviews. Word2Vec (Mikolov et al., 2013; Le and Mikolov, 2014) is a popular method to produce word embeddings (vector space model) from a corpus of text data. The Word2Vec word representation can be obtained by two alternative model architectures, namely CBOW or skip-gram. Unlike earlier literature, this dissertation thesis uses a Skip-Gram model for this task, which exploits word context more effectively and thus generates a more generalizable context when compared with the CBOW model (Mikolov et al., 2013). To train the Skip-Gram model, I use the hierarchical softmax algorithm, a computationally effective version of the softmax algorithm. To further enhance the detection performance, here I combine the generated word embeddings with bag-of-words in the first stage. In the second stage, to classify spam/legitimate messages, the proposed spam filtering model is trained using the ensemble learning algorithm with base learners represented by DFFNNs equipped with regularization techniques and rectified linear units.

This dissertation thesis aims to develop a new machine learning model based on DFFNN ensembles using a high-dimensional feature representation for spam filtering in diverse communication channels.

The remainder of this dissertation thesis is organized as follows. Chapter 1 reviews related work on filtering spam messages. Chapter 2 sets the objectives of this dissertation thesis. Chapter 3 introduces the proposed research methodology. Chapter 4 presents the datasets used for the experimental comparison and Chapter 5 introduces the strategies for data preprocessing and feature selection. Chapter 6 outlines the proposed spam filtering model and Chapter 7 briefly introduces the state-of-the-art models used for comparisons. Chapters 8 and 9 present the experimental settings and results, as well as a comparative analysis with the state-of-the-art methods used for spam filtering. Chapter 10 discusses the limitations and suggests possible future directions. Chapter 11 presents the theoretical and application contributions of this dissertation thesis and the last chapter concludes.

1. State-of-the-art in Spam Filtering

1.1 Importance of Spam Filtering

The idea of spam is very simple: to send a message to millions of people and profit from the one person who replies. Recent studies have shown that on average 80 % of e-mails is spam, with significant differences in spam rates among countries (see e.g. the Global Spam Map[1]). As a result, serious negative effects on the worldwide economy have been observed (Hoanca, 2006; Laorden et al., 2014; Obied and Alhajj, 2009), including lower productivity, the costs associated with delivering spam, and the cost with delivering spam and viruses/phishing attacks. Therefore, an effective spam filter may also improve user productivity and reduce the consumption of information technology resources such as the help desk. For individuals, more accurate spam filters may increase their trust in e-mail communication (Wei et al., 2008). The availability of unlimited pre-pay SMS packages has enabled the same approach for SMS spam. Increasing the cost of sending spam and reducing the burden spam places on users require highly accurate spam filters (Shen and Li, 2014).

Statistics show that a large proportion of all messages in social networks are spam messages. For instance, the study by Nexgate, a major company specialized in cyber security, reported that during the first half of 2013 there has been a 355% growth of social spam (Nexgate, 2013). For every seven new social media accounts, five new spammers are detected (Nexgate, 2013). The growing opportunities of social networks and their popularity have attracted many users. These days the base of social network users is steadily growing, and considerable amount of communication is done through social networks. However, along with legitimate and useful information, inappropriate and unwanted content is also released on these networks. Indeed, spam senders target social network users as well. Moreover, business social networks like LinkedIn are also affected (Statista, 2018b). This has serious economic and social consequences. Spam messages decrease work productivity, increase IT support related resources (help desk) and may even result in security incidents. This is why a considerable attention is given to spam filtering in social networks.

[1] https://globalsecuritymap.com

Fake reviews are unwanted and misleading reviews which can be submitted and listed on multiple online platforms, such as online shops and travel aggregators (Patel and Patel, 2018). In correlation with the number of internet users the number of users who shop online is growing as well. TripAdvisor is one of the most popular travel related website. User base of TripAdvisor is over 455 million average monthly unique visitors. Moreover, there are 600 million reviews about 7.5 million properties, restaurants, tours, etc. Many users take into consideration other users' reviews while choosing a property to stay. And fake review is becoming a problem due to the fact they may mislead potential buyers which will result in potential lawsuit against the seller and other adverse effects. Recent researches have shown that about every third review is fake on TripAdvisor (The Times, 2018). In order to guarantee fair competition, it is crucial to detect and remove fake reviews, since they give competitive advantage or disadvantage.

1.2 E-mail Spam Filtering

Spammers (persons sending spam messages) gather e-mail addresses from a wide range of sources, such as websites and chatrooms, send unsolicited messages in bulk. This has serious adverse effect on the recipient, including waste of time and resources. Specifically, e-mail spam has negative effects on the memory of e-mail server, CPU performance and user time. Moreover, the fraudulent practices of spammers may result in substantial financial losses of the recipients.

Although the global spam volume (percentage of total e-mail traffic) decreased to about 55% in the last decade (Statista, 2019a), the volume of e-mail messages with pernicious attachments (malware, ransomware, etc.) is steadily increasing (Dada et al., 2019). The largest share of spam e-mail spam was produced in China with about 20% of e-mail spam volume (Statista, 2019a).

Spam senders are strongly motivated to send bypass spam filters in order to increase the revenue. Therefore, spam filtering represents a challenging task because spammers use different techniques, in order to decrease spam detection rate. There are a number of methods such as using irrelevant, random or misspelled words, to evade commonly used spam filters.

Spam filtering techniques can be categorized into non-machine learning and machine learning approaches. The former include legislative approaches (Carpinter and Hunt, 2006; Talbot, 2008), changes to protocols and models of operation (Henning 2006), rule-, signature-, and hash-based filtering, whitelists (trusted senders) and blacklists, and traffic analysis (Caruana and

Li, 2008). Kaya and Ertugrul (2016) proposed an effective approach based on the probability of using characters in similar orders with respect to their UTF-8 values.

Machine learning spam filters automatically identify whether or not a message is spam based on its content (Fawcett, 2003). Following Sebastiani (2002) and Zhang et al. (2004), automated spam filtering can be defined as follows.

Let $D = \{d_1, d_2, \dots , d_i, \dots , d_N\}$ be a message set and $C = \{spam, legitimate\}$ be a class set. The task of a spam filter is to build a model to classify each message $d_i \in D$ as spam or legitimate. Misclassifying a legitimate message as spam (a false positive) and misclassifying spam as non-spam (a false negative) carries costs (Zheng et al., 2015). This is a challenging task because spammers usually attempt to decrease the probability their messages are detected as spam by using legitimate words (Shen, 2014).

With machine learning approaches, spam filtering starts with text pre-processing (Hagenau et al., 2013), with tokenization performed first to extract the words (multi-words) in each message. Next, typically, the initial set of words is reduced by stemming, lemmatization, and stop-words removal. Bag-of-words (BoW), also known as the vector-space model, is a common approach to represent the weights of the pre-processed words. Term frequency–inverse document frequency (*tf.idf*) is a popular specific weighting scheme. Feature selection algorithms, such as filters or wrappers (Almeida et al., 2011a; Liu et al., 2016; Trivedi and Dey, 2016a; Zhang et al., 2014), may then be applied to reduce the size of the feature space, which is useful mainly because not all classification methods can handle high-dimensional data. Finally, machine learning methods are applied to classify the preprocessed dataset.

The first spam classifiers employed NB algorithms due primarily to their simplicity and computational efficiency (Androutsopoulos et al., 2000; Metsis et al., 2006; Sahami et al., 1998). Concerning SVM, another popular spam-classification algorithm, it was shown that SVMs are robust to both different datasets and preprocessing techniques (Drucker et al., 1999). Its superiority to NB, *k*-nearest-neighbor (*k*-NN), decision trees, and NN approaches has been demonstrated in comparative studies (Lai, 2007; Vyas et al., 2015; Zhang et al., 2014). Artificial immune systems (AISs) (Watkins and Timmis, 2004) represent another promising method for spam filtering. Zitar and Hamdan (2013) used a genetic algorithm to train AISs to improve spam filter performance. Meta-learning algorithms (Garcia et al., 2010) have also recently attracted increasing attention (Trivedi and Dey, 2013). The combination of boosting and SVM outperformed single classifiers on several benchmark datasets in Trivedi and Dey (2016b). Similarly, boosting and bagging were reported to perform significantly better than NB and SVM in a stylometric spam filter (Shams and Mercer, 2016). Laorden et al. (2014) proposed an anomaly-based spam-filtering system that uses a data reduction algorithm on the labeled dataset, reducing processing time while maintaining high detection rates. Incremental training also reduces processing time (Sanghani and Kotecha, 2016). The above-mentioned classification methods usually require sufficient labeled data for the training process, data which are not always available in real-world applications. Semi-supervised approaches have therefore been employed to overcome this problem (Ahmed et al., 2015). Most recent reviews on e-mail spam filtering suggest that the future of e-mail spam filtering lies in content-based deep learning (Dada et al., 2019). Table 1 presents a summary of previous studies related to e-mail spam filtering.

Table 1: Summary of previous studies on e-mail spam filtering

Study	Method	Dataset (spam/legitimate)	Performance
Carpinter and Hunt (2006)	Heuristic filter + NB	SpamAssassin (2,399/6,953)	Acc=97.7%

Sculley and Wachman (2007)	SVM	Trec05 (52,790/39,399)
		Trec06 (24,912/12,910)
			AUC=0.991
			AUC=0.977

Mendez et al. (2007)	SVM	SpamAssassin (2,399/6,953)	Acc=98.5%
Fdez-Riverola et al. (2007)	Case-based Reasoning	SpamAssassin (4,150/2,801)	Acc=93.6%

Tzortzis and Likas (2007)	Deep Belief Networks	Enron1 (1,500/3,672)
		SpamAssassin (1,897/4,150)
			Acc=97.4%
			Acc=97.7%

Abi-Haidar and Rocha (2008)	AIS	Enron (1,000/1,000)	Acc=90.0%
Yu and Xu (2008)	SVM	SpamAssassin (2,222/2,777)	Acc=97.0%
Rozza et al. (2009)	Isotropic PCA	SpamAssassin (6,000/6,000)	Acc=98.9%
Zhou et al. (2010)	NB	UCI ML Repos. (1,813/2,788)	Acc=98.4%
Almeida el at. (2011b)	Multivariate Bernoulli NB	Enron1 (1,500/3,672)	Acc=94.8%
Liu and Wang (2012)	SVM	Trec07 (50,199/25,220)	AUC=0.992
Uysal and Gunal (2012)	Distinguishing FS	Enron (1,500/3,672)	Acc=94.4%
Almeida and Yamakami (2012)	MDL	Enron (17,171/16,545)	Acc=95.6%
Shams and Mercer (2013)	Bagged RF	Enron (17,171/16,545)	Acc=97.8%

Trivedi and Dey (2013)	Enhanced genetic programming	Enron (3,000/3,000)
		SpamAssassin (2,350/2,350)
			Acc=94.1%
			Acc=98.6%

Zitar and Hamdan (2013)	Genetic optimized AIS	SpamAssassin (580/420)	Acc=98.9%

Zhou et al. (2014)	NB	PU1 (481/618)
		Ling-Spam (481/2,412)
		UCI ML Repository (1,813/2,788)
			Acc=91.6%
			Acc=95.2%
			Acc=96.9%

Trivedi and Dey (2016a)	Relief + NB
	OneR + NB
		Enron (3,000/3,000)
		SpamAssassin (2,350/2,350)
			Acc=96.3%
			Acc=96.4%

Hassan (2016)	*k*-means + SVM	Enron (17,171/16,545)	Acc=97.4%
Chhogyal and Nayak (2016)	Natural language toolkit NB	Enron1 (1,500/3,672) Enron2 (1,496/4,361)	Acc=94.7%
Sanghani and Kotecha (2016)	Incremental SVM	Enron (17,171/16,545)	Acc=96.9%

Trivedi and Dey (2016b)	Boosted NB + SVM	Enron (3,000/3,000)
		SpamAssassin (2,350/2,350)
			Acc=95.6%
			Acc=98.6%

Fang (2016)	Maximum entropy + Incremental learning	SpamAssassin (250/220)	Acc=97.9%
Shams and Mercer (2016)	Natural language stylometry + Adaboost	SpamAssassin (1,884/4,149)	Acc=95.7%
George and Vinod (2018)	NB	Enron (1,500/3,672)	F-score=0.994

Gaurav et al. (2019)	RF	Enron (1,500/3,672)
		Ling-Spam (481/2,412)
			Acc=92.3%
			Acc=92.5%

Gupta et al. (2019)	Ensemble NB and DT	Enron (1,500/3,672)	Acc=92.4%
Diale et al. (2019)	SVM	Enron (17,171/16,545)	F-score=0.978

Legend: Acc – accuracy, AIS – artificial immune system, AUC – area under curve, DT – decision tree, FS – feature selection, MDL – minimum description length, NB – Naïve Bayes, PCA – Principal Component Analysis, RF – random forest, and SVM – support vector machine.

1.3 SMS Spam Filtering

Short message service (SMS) is a popular mean of communication these days. The increasing number of mobile phones in use leads to increased number of SMS sent and received. The rapid smartphones penetration has contributed to the growth of online instant messaging and SMS usage. According to Statista (2019b), the global smartphone penetration rate is projected to pass 40 percent for the first time. With 3.2 billion smartphone users worldwide and a global population of about 7.7 billion, the global smartphone penetration has reached 41.5 percent. Due to constant decrease of SMS price along with introduction of unlimited mobile phone plans, spammers can send spam messages at a very low cost or for free.

Various techniques were developed in order to address SMS classification. Hidalgo et al. (2006) benchmarked a set of classification algorithms and text representation methods in order to detect SMS spam messages. After evaluating results of the experiments, researches come to conclusion that that Bayesian filtering technique can be employed successfully to detect SMS Spam. While Healy et al. (2005) compared the performance of detecting SMS spam using another three popular machine learning classifiers, including *k*-NN, SVM and NB. The results of the experiments showed that SVM and NB demonstrated better classification performance than *k*-NN. Some other researches used terms normalization to create new attributes and later used to expand original text sampling aiming to alleviate factors which may lead to lower algorithm classification performance (Almeida et al., 2011). Another proposed method used distinctive features while eliminating uninformative ones considering certain requirements on term characteristics (Uysal et al., 2012). Indeed, SVM represents the most popular machine learning method in recent comparative studies (Kaliyar et al., 2018; Lee and Kang, 2019). Deep NNs (Gupta et al., 2018) and bio-inspired heuristic methods (Mokri et al., 2019) have also showed considerable improvement over traditional machine learning methods in recent SMS spam filtering studies, see Table 2 for an overview.

Table 2: Summary of previous studies on SMS spam filtering

Study	Classification method	Dataset (spam/legitimate)	Performance
Hidalgo et al. (2006)	SVM	SMS English (82/1,119)	AUC=0.930
Cormack et al. (2007)	Dynamic Markov Compression	SMS English (82/1,002)	AUC=0.988
Almeida et al. (2011)	SVM	UCI ML (747/4,827)	Acc=97.6%
Uysal and Gunal (2012)	Distinguishing FS	UCI ML (747/4,827)	Acc=97.4%
Uysal et al. (2012)	χ^2 filter + probabilistic classifier	UCI ML (747/4,827)	Acc=90.2%
Ahmed et al. (2015)	Apriori + ensemble learning	UCI ML (747/4,827)	Acc=96.2%
Chan et al. (2015)	SVM	UCI ML (747/4,827)	AUC=0.965
Najadat et al. (2016)	Discriminative multinomial NB	UCI ML (747/4,827)	Acc=96.5%
Almeida et al. (2016)	Markov Compression	UCI ML (747/4,827)	MCC=0.939
Aragao et al. (2016)	Factorial design SVM and NB	UCI ML (747/4,827)	Acc=99.4%
El Boujnouni (2017)	Support Vector Domain Description	UCI ML (747/4,827)	Acc=89.3%

Gupta et al. (2018)	CNN	UCI ML (747/4,827)
		Spam SMS 2011-12 (1,000/1,000)
			Acc=99.1%
			Acc=98.3%

Kaliyar et al. (2018)	SVM	UCI ML (747/4,827) SMS Assassin (2,123/2,195)	Acc=88.0%
Lee and Kang (2019)	SVM	SMS sentences (55,000/54,993)	Acc=95.7%
Mokri et al. (2019)	Octopod heuristic technique	UCI ML (747/4,827)	Acc=99.3%

Legend: Acc – accuracy, AUC – area under ROC curve, CNN – convolutional neural network, FS – feature selection, MCC – Matthews correlation coefficient, NB – Naïve Bayes, NN – neural network, and SVM - support vector machine.

1.4 Social Network Spam Filtering

User base of social networks is growing over the number of years. For instance, Facebook, one of the biggest social networks in the world, grew from one billion to two billion users just in 5 years (Statista, 2018a). Social network spam has become a major concern of industry and academia because it may include unwanted content, such as insults, hate speech, malicious links, etc. Such messages can be seen by the recipient’s followers. Moreover, they may lead to confusions and misdirection in public discussions (Zheng et al., 2015). Fighting social network spam with traditional legal methods has serious limitation because spam messages in social networks can be sent from different countries. It is important to note that spammers may use anonymizers, making it difficult to trace them. In order to overcome this problem, several social network spam filters have recently been developed (Adewole et al., 2017; Kaur et al., 2018). A list of related studies is showed in Table 3, presenting the methods and datasets used together with the resulting performance evaluation.

Features related to tweet content and user behavior were identified and used for machine learning using SVM (Benevenuto et al., 2010). Song et al. (2011) utilized relation features, such as the connectivity and distance between a tweet sender and receiver, to detect spam messages. A statistical analysis of language used in tweets represents an alternative approach (Martinez-Romo and Araujo, 2013), which identifies spam tweets in isolation (i.e., without user information) using their trending topics. Similarly, Antonakaki et al. (2016) exploited trending topics to detect spam campaigns in Twitter.

An SVM classifier was used by Lee and Kim (2013) to detect suspicious URLs in tweets. Their system makes use of correlated URL redirect chains extracted from tweets. URLs in social media have also been used in the behavior-based spam detection system proposed by Cao and Caverlee (2015). More precisely, the behavioral signals were obtained from both the URL sender and receiver. In other words, a high accuracy was achieved without using other tweets’ attributes such as those based on message content.

In addition to spam messages detection, recent studies have also considered an alternative task of social spammer (profile) detection. An NB classifier was proposed by Wang (2010) to detect spammers in Twitter. Gogoglou et al. (2016) identified the so-called “social bridges” to detect spammers in Twitter. These are reported as the major supporters of malicious users, and a graphtopology based classifier was used to detect such bridge linkages. A hybrid approach for identifying spam profiles was proposed by Aswani et al. (2018), combining social media analytics and firefly algorithm with chaotic maps for spam detection in Twitter marketing. A large Twitter dataset was used by Shen et al. (2017) to demonstrate that feature distributions between spammers and legitimate users are different. These feature distributions were used in a social spammer detection framework that integrated this information with a social regularization term incorporate into a classification model. Another way to tackle the issue of detecting spammers in Twitter was described by Bindu et al. (2018). A multilayer social network was defined, and the identification of spammers was based on the existence of overlapping community-based features of users represented in the form of hypergraphs, such as structural behavior and URL characteristics. A unified approach was proposed by Wu et al. (2016), utilizing the fact that social spammers tend to post more spam messages. Indeed, it was shown that combining social spammer filtering with spam message filtering improves the performance of both tasks.

Although Twitter represents the most frequently used source of data, alternative social networks have also been examined. For example, data from Sina Weibo were used to study features related to message content and user behavior (Zheng et al., 2015; Wu et al., 2016). The most important features were then used in the SVM classifier for spam detection. Extreme learning machines were used by Zheng e al. (2016) on a similar dataset. A semi-supervised social media spammer filtering approach was developed by Yu et al. (2017). This approach outperformed traditional supervised classifiers for the spammer detection task. Similar results were obtained for spam message detection in Hyves social network (Bosma et al., 2012). Bosma et al. (2012) introduced a framework for unsupervised spam detection in social networking sites, based on user spam reports. Using the same dataset, significant improvements were achieved by combining data oversampling with regularized deep neural networks (DNNs) (Barushka and Hajek, 2018a).

In recent years, there has been an increasing interest in dimensionality reduction techniques with the aim of improving the prediction performance and stability of social network spam filters (Al-Janabi et al., 2017). Several researchers employed feature selection and extraction methodologies to identify the most important features for social network spam filtering. The concept of rough set theory was applied by Dutta et al. (2018), concluding that the used methodology selected a smaller subset of features than those of the baseline methodologies (information gain, consistency subset evaluation, correlation-based feature selection, community detection and χ^2 evaluation). By considering important features of the posts and their corresponding comments, and finally applying the feature selection techniques, the method proposed by Sohrabi and Karimi (2018) selected the most effective features to detect spam using machine learning techniques. A probabilistic generative model (latent dirichlet allocation) was proposed by Song et al. (2017) to detect the latent semantics from user-generated comments. Incremental learning was then used to address the issue of the changing feature space. Three traditional feature selection methods were used by Al-Janabi et al. (2017), including information gain, Gini index and mean decrease accuracy. The latter measures attribute importance based on the accuracy of the random forest (RF) classifier. Evolutionary search algorithm was used in combination with χ^2 evaluation criterion by Adewole et al. (2019) to identify the reduced set of attributes for spam filtering in Twitter microblogging social network. Even better accuracy than the previously mentioned filter-based methods can be achieved using wrapper-based feature selection (Al-Zoubi et al., 2018). However, this approach is reportedly computationally intensive because the classifier must be trained on each feature subset. The main limitation of the wrapper-based approach proposed by Al-Zoubi et al. (2018) is the use of classification accuracy as the evaluation measure due to its unsuitability for different misclassification cost of spam and legitimate classes.

Regarding the classification methods used to categorize spam and legitimate messages (profiles), traditional machine learning methods have dominated in earlier research, such as NB, SVM and RF. To make use of unlabelled messages in the dataset, several studies have used methods with unsupervised learning in addition to supervised learning (Chen et al., 2017a; Sedhai et al., 2018). Ensemble-based approaches, such as Decorate (Lee et al., 2010) and Boosting (Lee et al., 2011), have been effectively used in a few studies, demonstrating that those methods can be more accurate in detecting spam than single classifiers. This can be attributed to the diversity of the base learners that reduces the problem of overfitting. However, the main limitation of the mentioned studies is the application of decision trees (DTs) as base learners, which suffer from several drawbacks, such as poor capacity to deal with high-dimensional datasets (Barushka and Hajek, 2018a).

Table 3: Summary of previous studies on social network spam filtering

Study	Classification method	Dataset (spam/legitimate)	Performance

Stringhini et al. (2010)	RF	Facebook profiles (173/827)
		Twitter profiles (500/500)
			FPR=0.020, FNR=0.010
			FPR=0.025, FNR=0.030

Lee et al. (2010)	Decorate	MySpace profiles (627/388)
		Twitter profiles (168/104)
			Acc=99.2%
			Acc=89.0%

Wang (2010)	NB	Twitter profiles (14/486)	F-score=0.917
Benevenuto et al. (2010)	SVM	Twitter messages (355/710)	Acc=87.2%
Lee et al. (2011)	Boosting RF	Twitter profiles (22,223/19,297)	Acc=98.4%
Jin et al. (2011)	Active learning	Facebook profiles	-
Thomas et al. (2011)	Suspension algor.	Twitter profiles (100/200)	-
Song et al. (2011)	LogitBoost, Bayes Net	Twitter messages (10 K/10 K)	TPR=0.997, FPR=0.006
Chu et al. (2012)	RF	Twitter campaigns (744/580)	Acc=94.5%
Bosma et al. (2012)	SSL	Hyves messages (698/497)	AUC=0.801
Yang et al. (2013)	RF	Twitter profiles (2,060/20,000)	F-score=0.900
Martinez-Romo and Araujo (2013)	SVM	Twitter messages (168 K/340 K)	F-score=0.883
Lee and Kim (2013)	SVM	Twitter messages (26,950/156,896)	Acc=91.9%
Bhat and Abulaish (2013)	ADTree	Facebook profiles (1,000/1,000)	AUC=0.985
Ahmed and Abulaish (2013)	NB, DT (J48)	Facebook profiles (165/155) and Twitter profiles (160/145)	Acc=95.7%
Miller et al. (2014)	DenStream+ *K*-means	Twitter profiles (208/3,031)	Acc=98.0%
Cao and Caverlee (2015)	RF	Twitter messages (124/214)	F-score=0.859, AUC=0.921
Zheng et al. (2015)	SVM	SinaWeibo profiles (11,488/17,646)	F-score=0.996
Antonakaki et al. (2016)	DT	Twitter (63,612/6.6 M)	TPR=0.810, FPR=0.006
Liu and Wang (2016)	ELM	Sina Weibo profiles (14,796/64,419)	F-score=0.996

Wu et al. (2016)	Co-detection of spammers and messages	Sina Weibo messages (25,681/27,803)
		Sina Weibo profiles (1,496/3,594)
			F-score=0.927
			F-score=0.795

Zheng et al. (2016)	ELM	Sina Weibo messages (500/500)	F-score=0.996
Song et al. (2017)	SVM	Youtube messages (210,283/845,092)	Acc=88.1% AUC=0.872
Soliman and Girdzijauskas (2016)	Unsupervised graph-based approach	Twitter profiles (2,072/17,322; 1,617/19,312; 3,109/12,128)	Acc=92.3%
Al-Janabi et al. (2017)	RF	Twitter messages (30 K/120 K)	AUC=0.920
Chen et al. (2017a)	RF+unsupervised learning	Twitter messages (1 M/ 1 M)	Acc=95.0%
Shen et al. (2017)	SVM	Twitter profiles (4,414/5,666)	F-score=0.879
Watcharenwong and Saikaew (2017)	RF	Facebook messages (600/600)	F-score=0.987
Yu et al. (2017)	SSL	Sina Weibo profiles (135/2,865)	F-score=0.920

Chen et al. (2017b)	RF	Twitter (9,945/90,055)	Acc=97.1%,
			F-score=0.838

Aswani et al. (2018)	*K*-Means+FA	Twitter profiles (4,923/9,312)	Acc=97.9%
Al-Zoubi et al. (2018)	SVM+WOA	Twitter profiles (204/196)	Acc=93.7%
Bindu et al. (2018)	Unsupervised SpamCom	Twitter profiles (22,223/19,276)	F-score=0.880
Dutta et al. (2018)	Graph-based greedy algorithm	Twitter messages (94 K/250 K)	Acc=81.0%
Sedhai and Sun (2018)	SSL	Twitter messages (49 K/22 K)	Acc=95.0%
Sohrabi and Karimi (2018)	DT	Facebook profiles (200 K)	Acc=92.0%

Barushka and Hajek (2018a)	DNN	Hyves messages (466/355)	Acc=92.8%,
			AUC=0.961

Adewole et al. (2019)	RF	Twitter messages (3,648/4,000)	Acc=93.2%,
			AUC=0.983

Legend: Acc – accuracy, AUC – area under ROC curve, DT – decision tree, ELM – Extreme learning machine, FA – firefly algorithm, FPR – false positive rate, FNR – false negative rate, NB – Naïve Bayes, RF – random forest, SSL – semi-supervised learning, SVM – support vector machine, TPR – true positive rate, and WOA – whale optimization algorithm.

1.5 Review Spam Filtering

Review spam (fake review) has been increasingly recognized as a major concern for online shopping. To affect consumers' decisions and thus achieve competitive advantage, positive and negative review spam are intended to promote or demote target products (Ren and Ji, 2017). As consumers have limited capacity to identify review spam (Harris, 2012; Heydari et al., 2015), machine learning methods have been employed for their early detection. To automatically classify reviews into spam or truthful class, an annotated corpus of reviews (with class labels) is typically used for training and testing. A considerable amount of literature has been published on the automatic detection of review spam in the last decade. A list of those studies is showed in Table 4, presenting the methods used, the datasets and the resulting performance evaluation.

Jindal and Liu (2007) presented the first study aimed to detect product review spam based on the similarity of review and product features. More precisely, spammers' tendency to duplicate their product reviews was utilized. Motivated by this early effort, the studies that followed developed review spam detection systems using the cosine similarity between reviews (Lim et al., 2010; Li et al., 2011). To detect spammers who can adapt their behavior, Wang et al. (2011) proposed a heterogeneous review graph that captures the relationships among reviews, reviewers and reviewed shops. Thus, the trustiness of reviewers, the honesty of reviews and the reliability of shops could be calculated without considering review content. Inspired by this approach, Liu et al. (2019) proposed a probabilistic graph classifier, in which the multimodal embedded representation of nodes is obtained using a bidirectional NN with attention mechanism. In contrast, Lau et al. (2011) developed a review spam detection approach based on text mining only. Several types of features were used by Li et al. (2011), including review content, its sentiment, product features and user profile, to classify review spam using semi-supervised machine learning methods. Review metadata (content, timestamp and rating) were combined with relational data in a unified semi-supervised framework called SpEagle (Rayana and Akoglu, 2015). Ghai et al. (2019) show that the rating deviation of a particular review from others indicate review spam. Spam attacks were reported to be correlated to review ratings and, therefore, abnormal temporal patterns in the ratings may indicate spam attacks (Xie et al., 2012). By elaborating this idea, a list of indicative signals of review spam over time was used for real-time detection of abnormal review events (Ye et al., 2016; Li et al., 2017b). Furthermore, temporal features were combined with users' spatial patterns to find that review spam exhibit geographical outsourcing and spammers are more active in weekdays (Li et al., 2015). A rule-based feature weighting scheme was proposed by Asghar et al. (2020) to combine review-based, reviewer-based and product-based features.

Most existing review spam detection systems extract informative features from the review content. Such features are typically represented by bag-of-words (*n*-grams) (Ott et al., 2012; Ott et al., 2013), psycholinguistic word lists (e.g., positive/negative words or spatial words) (Li et al., 2014) or part-of-speech tagging (e.g., first-person pronouns) (Li et al., 2017a). Aspect sentiment was identified in Liu et al. (2018) to detect fraud users. Xue et al. (2019) integrated the deviation of user's aspect sentiment into a framework calculating the trust scores for users, reviews and products, respectively. Word embeddings have recently been used to obtain the semantic representation of reviews. Ren and Ji (2017) proposed the pre-trained CBOW model tuned on actual review datasets using convolutional neural network (CNN) to improve the detection accuracy. The CBOW model was also used together with relational features to develop a semi-supervised framework in Yilmaz and Durahim (2018). Word embeddings were also trained using sentence-based CNNs to produce document representations for review spam detection in several product domains (Li et al., 2017a).

Table 4: Summary of previous studies on review spam filtering

Study	Classification method	Dataset (spam/legitimate)	Performance
Jindal and Liu (2007)	LR	Amazon (5.8 M reviews)	AUC=0.780
Li et al. (2011)	NB, Co-training	Epinions (1,398/4,602)	F-score=0.631
Chandy and Gu (2012)	DT, LCGM	App Store (6.3 M)	Acc=73.6%
Ott et al. (2013)	SVM	Hotels (800/800)	Acc=86.0%
Shojaee et al. (2013)	SVM	Hotels (800/800)	F-score=0.840
Mukherjee et al. (2013)	SVM	Yelp hotels and restaurants (802/4,876 and 8 K/50 K)	Acc=86.1%
Li et al. (2014)	SAGE	Hotels (1080/800) Restaurant (320/400) and doctors (232/200)	Acc=64.7%
Li et al. (2015)	SVM	Restaurants (6.1 M in total)	Acc=85.0%
Rayana and Akoglu (2015)	SSL	Yelp (80,456/528,141)	AUC=0.794
Sun et al. (2016)	Bagging	Products (800/1200)	F-score=0.772
Li et al. (2017a)	CNN, SWNN	Hotels (800/800), restaurants (200/200) and doctors (356/200)	Acc=83.5%
Ren and Ji (2017)	CNN, GRNN	Hotels (800/800), restaurants (200/400) and doctors (200/200)	Acc=83.5%
Elmurngi and Gherbi (2017)	*k*-NN, NB, DT, SVM	Movies (2,000 in total)	Acc=81.8%
Rout et al. (2017)	*k*-NN, RF	Hotels (800/800)	Acc=77.5%
Yilmaz and Durahim (2018)	SSL	Yelp (80,456/528,141)	AUC=0.832
Ahmed et al. (2018)	SVM	Hotels (800/800)	Acc=90.0%
Zeng et al. (2019)	LSTM ensemble	Hotels (800/800), restaurants (200/200) and doctors (356/200)	Acc=83.4%
Barbado et al. (2019)	AdaBoost	Yelp (9,456/9,456)	F-score=0.810
Kennedy et al. (2019)	BERT	Hotels (800/800) and Yelp (78,346/ 78,346)	Acc=89.1%
Liu et al. (2019)	LR	Dianping restaurants and hotels (31 K and 98 K in total)	F-score=0.810

Legend: Acc – accuracy, AUC – area under ROC curve, BERT – bidirectional encoder representations from transformers, CNN – convolutional neural network, DFFNN – deep feed-forward neural network, DT – decision tree, FNR – false negative rate, FPR – false positive rate, GRNN – general regression neural network, *k*-NN – k-nearest neighbor, LCGM – latent class graphical model, LDA – latent dirichlet allocation, LIWC – linguistic inquiry and word count, LR – logistic regression, LSTM – long short term memory, NB – Naïve Bayes, POS – part-of-speech tagging, RF – random forest, SAGE – sparse additive generative model, SSL – semi-supervised learning, SVM – support vector machine, SWNN – sentence weighted neural network.

Regarding the classification methods used to detect spam and truthful reviews, machine learning methods have dominated in earlier research. Logistic regression (LR) has been first employed as the traditional machine learning method owing to its capacity to produce the probability estimate reflecting the likelihood that a review is a review spam (Jindal and Liu, 2007). However, traditional machine learning methods, such as LR and *k*-NN, may suffer from at least two drawbacks (Barushka and Hajek, 2018b). First, these methods are not effective in handling high dimensional review spam data. This is important because a large number of word features is usually present in these data. Second, those methods cannot deal with data sparsity effectively. This is critical because each review usually contains only a small number of words or phrases. To overcome these problems, other machine learning methods became popular for review spam detection, such as NB (Li et al., 2011) or SVM (Mukherjee et al., 2013; Li et al., 2015). Similarly, evolutionary algorithms (Pandey and Rajpoot, 2019) and ensemble learning methods (Rout et al., 2017; Barbado et al., 2019) have been utilized to overcome the problems of convergence and overfitting, respectively. A detailed survey of the traditional machine learning methods used to detect fake review was carried out by Crawford et al. (2015), Patel and Patel (2018) and Vidanagama et al. (2020).

Recent advances in this automatic fake review detection suggest that more complex features can be extracted from the high-dimensional data using DNNs. Therefore, spam filtering models using DNNs such as general regression neural network (GRNN) (Ren and Ji 2017), generative adversarial network (GAN) (Tang et al., 2019), CNN (Li et al., 2017a), DFFNN (Barushka and Hajek, 2019b) and long short term memory (LSTM) (Zeng et al., 2019) have gained much attention in recent years.

1.6 Partial Conclusion

In summary, previous related literature attempted to overcome the problem of high-dimensional data (the curse of dimensionality) by selecting the most important features, regardless of whether content-based features or user behavior features. This was mainly due to the risk of overfitting or poor convergence of the used classification methods. However, useful information may be hidden in higher-order features that can be extracted by using deep NNs (Barushka and Hajek, 2016). In fact, additional hidden layers enable the recombination of features and thus to capture higher complexity and abstraction in high-dimensional datasets (Hinton et al., 2012).

Moreover, ensemble methods have become popular in spam detection tasks due to their capacity to reduce the risk of overfitting and variance (Kaur et al., 2018). In order to take advantage of these approaches, this dissertation thesis uses DFFNNs as base learners in several ensemble learning schemes, including Boosting, Bagging and Random subspace.

2. Aim and Objectives of the Dissertation

The aim of the thesis is to propose a spam filtering model that integrates a high-dimensional feature selection and a regularized DFFNN model with rectified linear units to capture complex features from the high-dimensional data.

To achieve this aim, the following specific objectives are defined:

- Collect and preprocess spam datasets. Seven benchmark spam datasets are used for spam filtering. Specifically, e-mail datasets (both personalized and non-personalized), SMS dataset, social network datasets and review datasets are included to ensure that the proposed model can be applied across different electronic spam domains. Thus, testing different spam datasets enables me to demonstrate the robustness of the proposed spam filter. To preprocess the datasets, all words will be converted to lower-case letters and tokenization will be performed. To represent tokens, *n*-grams (unigrams, bigrams and trigrams) will be used. Furthermore, stop-words will be removed to avoid noise in the data.
- Perform high-dimensional feature selection. To represent the weights of the pre-processed words, the *tf.idf* scheme will be used as the most common BoW approach. Unlike raw term frequency, *tf.idf* considers both term rareness and document length. To select the most relevant words, the terms will be ranked according to their *tf.idf* weights. For the experiments the top 100, 200, 1000 and 2000 words in a BoW fashion will be used. Using too many features in a spam filter may not only extend computation time but also deteriorate classification performance due to the higher complexity. Therefore, the use of various numbers of top *n*-grams may also be considered a feature selection method in spam filtering. Moreover, in order to consider word context, the Skip-Gram word embedding model will be utilized to build a vector model so that words or phrases are mapped from the vocabulary to vectors of numerical values.
- Propose a regularized DFFNN model with rectified linear units for spam filtering. This model will be further enhanced with ensemble learning and inclusion of word embedding preprocessing. Complex tasks require many hidden units to model them accurately. DNNs with many parameters are extremely powerful machine learning systems that contain multiple hidden layers to process complicated relationships

between inputs and outputs. However, complex adaptation to training data may lead to overfitting, preventing high accuracy on testing data. Overfitting can be effectively addressed through dropout regularization, in which the units (hidden and visible) in a NN are temporarily removed from the network. Moreover, commonly used sigmoidal units reportedly suffer from slow convergence of optimization to a poor local minimum. Rectified linear (ReL) units tackle this problem. In order to further improve the accuracy rate, ensemble learning techniques with regularized DNNs as base learners will be utilized. It is assumed that this approach will lead to better generalizability and robustness compared with single estimators.

- Benchmark the proposed spam filtering model against other existing models in terms of the following prediction measures: accuracy (Acc), area under receiver operating characteristic (AUC) curve, false negative rate (FNR), false positive rate (FPR), and F-score. Moreover, different misclassification cost ratios will be considered. To demonstrate the effectiveness of the proposed spam filtering model, the results will be compared with the state-of-the-art machine learning approaches to spam filtering based on supervised learning, such as factorial design SVM (Aragão et al., 2016), incremental C4.5 DT (Sheu et al., 2017), RF (Khorshidpour et al., 2017) and CNN (Li et al., 2016). In addition, several machine learning methods, such as *k*-NN, AdaBoost and Bagging, will be used to represent traditional spam filtering methods.

3. Research Methodology

The research methodology of the dissertation is depicted in Fig. 1. First, datasets from several application domains will be collected, including benchmark datasets on e-mail, SMS, social networks and online reviews. Then, text preprocessing will be performed to remove inconsistencies and noise in the datasets. In the third step, features will be selected using two schemes, *n*-grams based on their *tf-idf* weights and word embeddings using the Skip-Gram model. The experiments will be performed on training and testing datasets using 10-fold cross-validation to ensure the reliability of the results. Different machine learning algorithms will be proposed to train the spam filtering models. First, single regularized DFFNN with ReL will be examined. Further, multiple DFFNNs will be used in several ensemble-based learning modes.

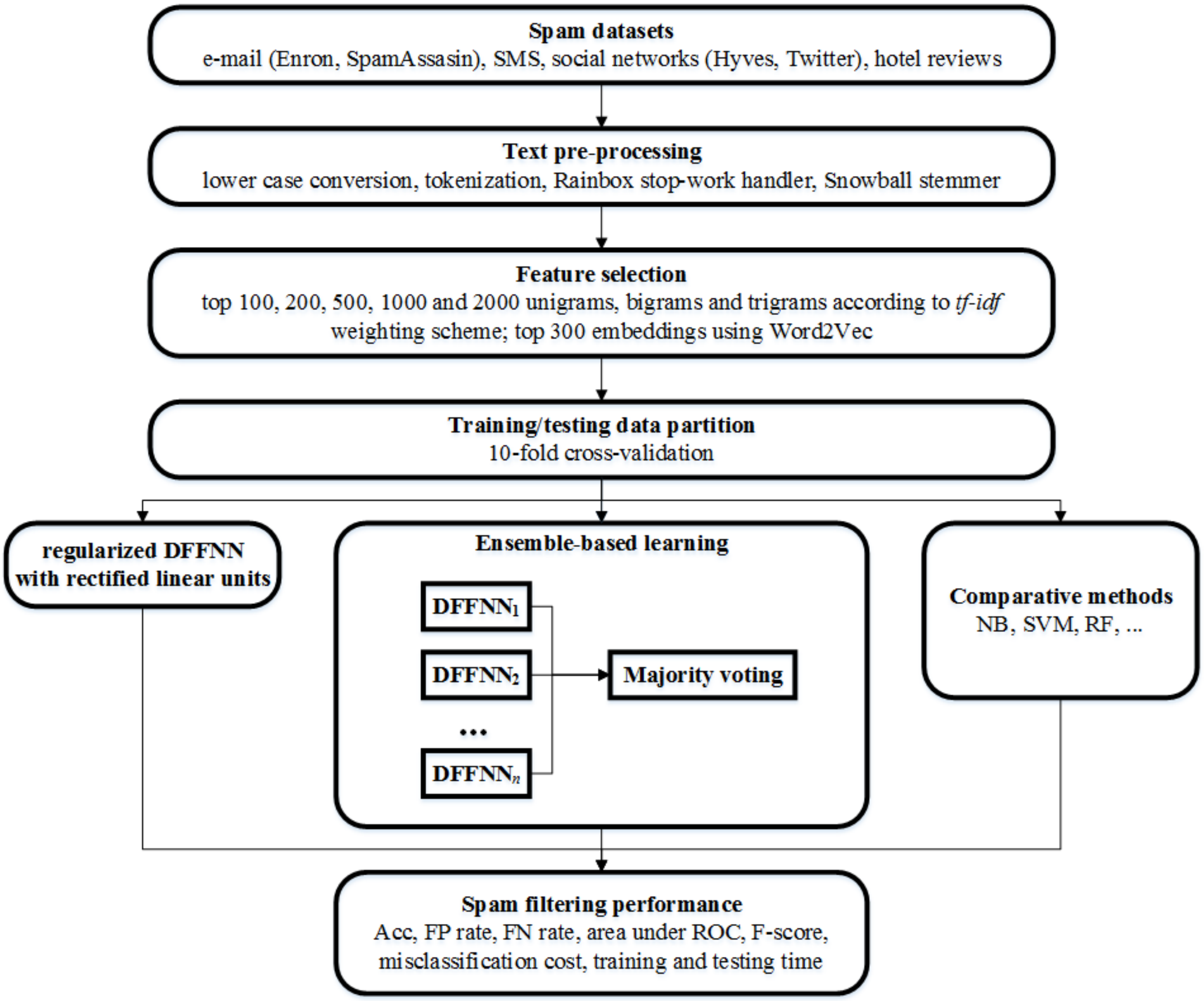
Spam datasets
e-mail (Enron, SpamAssasin), SMS, social networks (Hyves, Twitter), hotel reviews
Text pre-processing
lower case conversion, tokenization, Rainbox stop-work handler, Snowball stemmer
Feature selection
top 100, 200, 500, 1000 and 2000 unigrams, bigrams and trigrams according to tf-idf weighting scheme; top 300 embeddings using Word2Vec
Training/testing data partition
10-fold cross-validation
regularized DFFNN with rectified linear units
Ensemble-based learning
DFFNN1
DFFNN2
...
DFFNNn
Majority voting
Comparative methods
NB, SVM, RF, ...
Spam filtering performance
Acc, FP rate, FN rate, area under ROC, F-score, misclassification cost, training and testing time

Figure 1: Research methodology

The results will be compared with several existing machine learning-based spam filtering methods. For the comparative analysis, several evaluation measures will be used to ensure that the proposed model performs well on both spam and legitimate classes. In addition, to consider the greater importance of classification performance on the legitimate class, experiments will be performed using different misclassification cost ratios. Finally, training and testing times will be measured to evaluate computational effectiveness of the proposed models. To show the statistical differences, non-parametric statistical tests will be performed across all the datasets in the last step.

4. Datasets

When evaluating the performance of different spam filters, several benchmark datasets are usually employed. It is crucial to choose datasets from different application domains to prove that the proposed spam filtering model is widely applicable. There are four classes of datasets used in this dissertation thesis, namely as e-mail, SMS, social network and online review datasets. In addition to the communication channel variety, it is important to examine whether the proposed model performs well in different data environments, such as the level of data sparsity and class imbalance.

In order to measure the performance of the proposed model against existing models, the following publicly available spam datasets were used:

1) Enron[2],
2) SpamAssassin[3],
3) SMS[4],
4) Hyves, the Dutch social networking site[5],
5) Twitter[6],
6) Positive hotel reviews and
7) Negative hotel reviews[7].

The Enron spam dataset (Méndez et al., 2007) is a popular personalized dataset with spam and ham e-mail messages. This spam dataset has been used in a number of studies, see Guzella and Caminhas (2009) for an overview. This dataset, also called Enron 1, contains a total of 5,172 e-mails, including 3,672 legitimate and 1,500 spam e-mails. The original forms of messages are used, this is in non-Latin encodings with several slight modifications (legitimate e-mails sent by the owners of the mailboxes to themselves and a handful of virus-infected e-mails are removed). Each message is in a separate text file.

[2] http://csmining.org/index.php/enron-spam-datasets.html
[3] http://csmining.org/index.php/spam-assassin-datasets.html
[4] https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection
[5] http://ilps.science.uva.nl/framework-unsupervised-spam-detection-social-networking-sites/
[6] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5549928/bin/pone.0182487.s003.xlsx
[7] http://myleott.com/op-spam.html

The SpamAssassin dataset (Henning, 2006) is another popular corpus which has been used as a benchmark in many studies. This dataset contains 2,798 e-mails, of which 1,401 are legitimate and 1,397 are spam e-mails. This dataset is composed of randomly collected e-mails over a given time period and it is therefore suitable for testing non-personalized spam filters (Shams and Mercer, 2016).

A SMS spam dataset (Almeida et al., 2011) was chosen in order to diversify spam corpora. Unlike Enron and SpamAssasin datasets, SMS spam dataset includes 4,827 legitimate and 747 spam SMS messages, this is a total of 5,574 messages. The sources used in this corpus were the Grumbletext Web site (425 SMS spam messages), the NUS SMS Corpus (3,375 legitimate SMS), 450 legitimate SMS messages collected from Caroline Tag's PhD Thesis and the SMS Spam Corpus v.0.1 Big (1,002 legitimate SMS ham messages and 322 spam messages), see Almeida et al. (2011) for details. The average number of tokens in legitimate SMS is 13.18 while 23.48 in spam SMS.

The Hyves social network dataset contained both labelled and unlabelled messages from Hyves, the Dutch social networking site (Bosma et al., 2012). As a supervised learning approach is used in this thesis, the unlabelled (unannotated) messages were excluded from the dataset. Unsolicited and promotional messages were labelled as spam. Most of these messages were non-commercial spam messages, such as friend and group invitations or requests to follow a user on Twitter. The dataset includes the following types of information: message content, spam report and user information. The Hyves social network spam dataset contained 466 spam messages and 355 legitimate messages. The messages were represented as the arrays of json objects with the following fields: the annotation of the object (either spam or legitimate), anonymized IDs of the reporters of the message, anonymized ID of the author of the message, and bag of words representation of the message (an anonymized ID was assigned to each word). Similarly to SMS spam, messages in social networks are generally short, corresponding to sparser datasets. The average legitimate message had 33.15 tokens while the average spam message had 34.70 tokens.

The Twitter dataset was originally used by Chen et al. (2017). Unlike the Hyves dataset, the Twitter dataset is highly imbalanced. The original dataset had tweet ID and label only. The authors labelled the dataset manually and provided the links to the used tweets along with their labels. Therefore, the content of messages can be retrieved using API. I attempted to download all the tweets in July 2018. However, many messages were filtered and removed by that time. As a result, the final dataset consisted of 61,675 tweets, 4,198 of them labelled as spam and 57,476 as ham.

The Hotel review datasets consist of positive and negative reviews. Both datasets were provided by the Cornell University. The positive hotel review spam dataset contained 400 legitimate and 400 spam positive reviews from TripAdvisor (20 legitimate and 20 spam reviews for each of the 20 selected hotels) (Ott et al., 2012). The spam reviews were gathered using Amazon Mechanical Turk. Only a single review per Turker was allowed, and unreasonably short or plagiarized reviews were rejected. For the positive dataset, only 5-star reviews were included.

A similar procedure was used to collect the negative hotel review spam dataset (Ott et al., 2013). Again, Turkers were employed to provide spam reviews on 20 popular hotels, such as such as Affinia Chicago or Ambassador East Hotel, and corresponding legitimate reviews were obtained from several online review communities, such as Expedia, TripAdvisor or Hotels.com. For the negative dataset, only 1- or 2-star reviews were used. The average review length for both datasets was 116 words. The datasets included the following types of information: message content, spam label, hotel information, polarity of the message, and travel agency aggregator name.

The Enron dataset consists of 5,171 messages and the dataset is relatively balanced with about 29% of spam messages. The SpamAssassin dataset is almost perfectly balanced and has 2,798 messages. Unlike the e-mail datasets, the SMS and Twitter datasets are highly imbalanced, including 15.4 % and 7.3 % spam messages, respectively. The Social network and Hotel review (both polarity) datasets are well balanced and relatively small in size (less than 900 messages). In contrast, the Twitter dataset is the largest dataset with more than 60,000 messages. The results in Table 5 also demonstrate that the e-mail and review datasets tend to be longer than the social network and SMS messages, indicating higher data sparsity of the latter ones. Moreover, the negative hotel review messages are longer than the positive ones.

Table 5: Datasets

dataset	spam / legitimate	average message length (# words)
Enron	1,499 / 3,672	189.2
SpamAssassin	1,397 / 1,401	117.4
SMS	748 / 4,849	15.6
Hyves social network	466 / 355	37.8
Twitter	4,198 / 57,476	17.7
Hotel review (positive)	400 / 400	119.4
Hotel review (negative)	400 / 400	178.1

5. Data Preprocessing and Feature Selection

Features used for detecting spam messages can generally be categorized into those related to the characteristics of senders (sender-centric features) and those associated with the content of messages (message-centric features) (Crawford et al., 2015). As the latter approach has been considered more effective in previous studies, here I focus on how text preprocessing of messages affects the performance of automated methods for spam detection.

5.1 Data Preprocessing and Feature Selection Methods

A large number of features can be extracted from the text of consumer reviews, including bag of words, term frequencies, part of speech (grammatical tagging) or semantic features (Heydari et al., 2015). In the BoW approach, the presence/absence of individual words (or adjacent words) represents the features. In other words, word frequencies are not taken into consideration in this approach. To give different weights to words with different count of occurrences, term frequencies can be calculated. The semantic features represent the underlying meaning of words.

Before extracting the above-mentioned features, several text preprocessing strategies can be applied to improve text mining effectiveness. Tokenization, stop words removal and stemming have been considered particularly important (Uysal and Gunal, 2014). Tokenization transforms the text content into individual words/word phrases. To reduce the dimensionality of term space, the most common words (so-called stop words), such as articles and prepositions, can be removed. Word roots are identified in the process of stemming and, thus, similar to stop words removal, the dimensionality of term space is reduced.

Harris et al. (2012) used a popular QuickLM language model compiler to produce unigram (individual words) models for both high-rated and low-rated pooled review sets. All the words were transformed to lower cases and no stemming was performed. Natural Language Toolkit is another commonly used tool for fake review preprocessing, including tokenization and stemming Liu and Pang (2018). Unigram and bigram *tf* (term frequency) model was used by (Ott et al., 2013) to detect fake reviews in two datasets, namely positive and negative deceptive opinion datasets. It was shown that the *n*-gram based SVM classifier significantly outperformed human judges.

A more detailed analysis of text preprocessing techniques was performed by (Ahmed et al., 2018) who proposed an *n*-gram (*n* = 1, 2, 3 and 4) language model to feed six different machine learning methods. To preprocess the text data, stop words were first removed to reduce noise caused by irrelevant words. Then, the Porter stemmer was applied and the words were selected according to their *tf* and *tf.idf* (term frequency-inverted document frequency), respectively. SVM performed best among the tested machine learning methods, with the highest accuracy achieved for highly dimensional unigram and bigram language models. Moreover, *tf.idf* weighting scheme was more effective than the *tf* approach. A different approach to select the most important features was chosen by (Li et al., 2011). The top 100 unigrams and bigrams were selected based on the value of the χ^2 statistic. The weights were then normalized by the length of the review. Similarly, Kullback-Leibler-divergence was used as a weighting scheme to select the words for a sentence weighted NN classifier in (Li et al., 2017a). Unigrams, bigrams, and trigrams were also recently used to obtain sentence representations based on DNNs (Ren et al., 2017). Sun et al. (2013) developed a product word composition model based on CNNs to incorporate product-review relations. An improved performance was then achieved in combination with SVM bigram and trigram classifiers. Word context was considered by (Barushka and Hajek, 2019a) in an integrated DNN model combining BoW and word embeddings. Multimodal embedded representation of reviews, authors and products was used by Liu et al. (2019) to perform fake consumer review classification in a large context.

As pointed out above, so far, there have been a number of results focusing on content-based detection of spam messages. However, to the best of the author’s knowledge, up until now, there has been no research on the role of text processing techniques over multiple spam detection domains.

To preprocess the textual data, a number of techniques were applied. Here I provide their brief description. First, sentence tokenizer was used to split the texts of the reviews into sentences. Second, the tokenization of words was performed using the *n*-gram tokenizer. In this step, sentences were split into words or word segments (phrases) using the following delimiters: .,;:'"()?!. I also examined the effect of the lengths of word segments on classification performance. More precisely, unigrams (*n* = 1), bigrams (*n* = 2) and trigrams (*n* = 3) were extracted. Furthermore, I considered the removal of stop words (the most common words in a language and have limited linguistic meaning) using the Rainbow stopword list, and stemming using the Snowball stemmer (words were reduced to a root by removing inflection through dropping unnecessary characters). Again, we tested the effect of stopword removal and stemming on the results of classification. Also note that all words were first transformed to lowercase letters. The effect of data dimensionality was also taken into consideration. Different numbers of selected features were considered, namely 100, 200, 500, 1000 and 2000 features. The features were selected according to their weights. Appropriate weighting scheme must be chosen for that purpose. In this work, I considered two common weighting schemes, binary and non-binary. In the binary weighting scheme, w_{ij}=0 and w_{ij}=1 for the *i*-th word and *j*-th message indicate absence and presence of a word, respectively. Word counts are taken into account in the non-binary weighting schemes. The *tf.idf* weighting scheme is a commonly used approach. Term frequency *tf* denotes the number of word occurrences, while *idf* informs about the distribution of the *i*-th word in all reviews (content-bearing words are rarer). The weight w_{ij} can be calculated as follows:

$$w_{ij} = (1+log(tf_{ij}))\times idf_i, \text{ where} \qquad (1)$$

$$idf_i = log(N/df_i), \qquad (2)$$

where df_i is document frequency of the *i*-th word and *N* denotes the number of reviews. Finally, review lengths were considered using the normalization of *tf*.

In order to benchmark the preprocessed datasets, various classification methods were applied, including traditional classification methods used in earlier related research, namely NB (Elmurngi and Gherbi, 2017), SVM (Ott et al., 2013) and NN (Barushka and Hajek, 2019a). SVM was trained using the sequential minimal optimization (SMO) algorithm with polynomial kernel function and different settings of the complexity parameter $C = \{2^0, 2^1, ... , 2^6\}$ was examined. For NN, a multi-layer neural network with dropout and one hidden layer with different numbers of neurons {10, 20, 50, 100, 200} was tested. Note that in the following section, the results are reported as obtained for the optimum setting of the classification methods. To provide a reliable empirical evidence, the 10-fold cross-validation procedure was applied to the datasets. Thus, the results for 10 testing runs were obtained and, hereinafter, I report the average performance for all the methods. The performance was measured using two standard metrics applied to text classification, Acc, AUC and F-score. Acc is the percentage of correctly classified messages. AUC measures the trade-off between the percentage of correctly classified spam messages and the percentage of incorrectly classified legitimate messages at various threshold values. F-score evaluates the balance between precision and recall measures. Precision is the ratio between correctly classified fake reviews and all messages classified as spam. Recall is defined as the percentage of correctly classified spam messages.

5.2 Experimental Results on Preprocessing Strategies

To perform the experiments, I started with the definition of baseline setting. For this, the setting used in a recent study was adopted (Barushka and Hajek, 2019a). In this setting, 2,000 words (trigrams) were extracted using the *tf.idf* weighting scheme with stopword removal, stemming and document normalization. Then, the effects of the text preprocessing techniques can be examined using this baseline.

Tables 6-9 show the accuracy of the tested methods obtained for the baseline setting and the effects of different text preprocessing techniques. The results demonstrate that adding more features improves classification accuracy for all the datasets. However, using more than 1,000 words may decrease performance for certain algorithms and dataset. This finding also suggests that the used methods are effective in tackling high-dimensional datasets and that feature reduction is not necessary for this task.

Table 6: Accuracy obtained for different text preprocessing strategies for e-mail datasets

	Enron			SpamAssassin		
Method	NB	SVM	NN	NB	SVM	NN
baseline	69.27	97.99	98.59	95.10	99.39	99.14
100 words	87.68	94.72	92.69	93.32	97.03	96.03
200 words	91.61	95.90	95.53	94.18	98.25	97.32
500 words	89.07	96.44	97.64	95.68	98.86	98.57
1,000 words	80.64	97.25	98.32	95.93	99.39	97.71
unigrams	88.28	98.09	98.67	95.21	99.43	99.29
bigrams	78.20	98.01	98.47	97.07	99.46	99.07
binary weights	70.99	97.91	98.67	95.96	99.32	99.11
no stemming	69.27	97.99	98.47	96.32	99.39	99.04
no stopword removal	71.57	98.22	98.40	95.89	99.43	99.14

Note: the results better than baseline are underlined

Second observation is that the use of unigrams is not sufficient and higher accuracy can be achieved by using bigrams or trigrams. Third, the binary weighting scheme had no consistent impact on the accuracy. While binary weights improves accuracy for social network dataset, however it decreases accuracy for positive dataset. The results demonstrate that stemming also help slightly to improve accuracy rate, while stop words have little impact on performance.

Table 7: Accuracy obtained for different text preprocessing strategies for SMS dataset

	SMS		
Method	NB	SVM	NN
baseline	76.83	98.18	98.61
100 words	95.21	96.75	96.96
200 words	95.59	97.41	97.68
500 words	53.14	96.19	98.11
1,000 words	54.80	95.94	98.36
unigrams	55.55	97.57	97.55
bigrams	55.90	98.27	98.64
binary weights	96.43	98.02	98.59
no stemming	76.83	98.18	98.55
no stopword removal	71.18	98.39	98.62

Note: the results better than baseline are underlined

Table 8: Accuracy obtained for different text preprocessing strategies for social network datasets

	Hyves			Twitter		
Method	NB	SVM	NN	NB	SVM	NN
baseline	64.43	90.74	91.84	93.89	78.87	91.87
100 words	88.31	85.14	90.99	82.12	83.22	85.13
200 words	82.10	84.53	88.22	88.18	85.23	86.77
500 words	62.12	86.00	90.50	93.10	86.35	89.30
1,000 words	64.79	89.16	91.72	93.38	83.31	87.81
unigrams	83.80	79.04	92.81	92.92	81.44	81.69
bigrams	72.71	90.50	91.35	95.01	79.64	91.38
binary weights	67.23	90.99	92.69	93.92	81.19	89.96
no stemming	64.43	90.74	92.33	93.89	78.87	91.52
no stopword removal	64.43	90.74	92.08	94.05	76.72	92.40

Note: the results better than baseline are underlined

Table 9: Accuracy obtained for different text preprocessing strategies for hotel review datasets

	Positive hotel reviews			Negative hotel reviews		
Method	NB	SVM	NN	NB	SVM	NN
baseline	85.75	88.00	89.87	86.00	86.50	88.87
100 words	75.50	79.00	77.12	75.62	76.50	73.75
200 words	81.25	81.00	83.25	77.75	79.00	81.50
500 words	83.62	81.87	87.75	82.00	81.37	86.50
1,000 words	85.00	85.50	87.87	81.62	84.50	87.62
unigrams	83.12	86.75	88.37	78.50	85.50	88.12
bigrams	86.25	88.00	90.25	85.62	86.25	89.37
binary weights	82.25	84.12	86.25	86.12	86.12	88.62
no stemming	85.75	88.00	89.75	86.00	86.50	89.25
no stopword removal	86.00	87.87	89.62	84.25	87.00	90.00

Note: the results better than baseline are underlined

As presented in Tables 10-13, we can observe an increase in F-score performance with the increase in the number of features. The impact of binary weights was particularly positive for the NB classifier. Furthermore, bigrams and trigrams worked better than unigrams. Bigrams and trigrams dominated depending on the algorithm and dataset. Stemming helps increase the F-score measure in most experiments. Moreover, removing stopwords also slightly improved the results. Overall, the results for Acc and F-score demonstrate similar patterns.

Table 10: F-score for different text preprocessing strategies for e-mail datasets

	Enron			SpamAssassin		
Method	NB	SVM	NN	NB	SVM	NN
baseline	0.724	0.986	0.990	0.952	0.994	0.991
100 words	0.905	0.962	0.948	0.936	0.964	0.961
200 words	0.937	0.971	0.968	0.943	0.970	0.974
500 words	0.917	0.975	0.983	0.958	0.983	0.986
1,000 words	0.843	0.981	0.988	0.960	0.989	0.977
unigrams	0.911	0.986	0.991	0.951	0.995	0.993
bigrams	0.819	0.986	0.989	0.971	0.994	0.991
binary weights	0.744	0.985	0.991	0.959	0.995	0.991
no stemming	0.724	0.986	0.989	0.963	0.994	0.990
no stopword removal	0.750	0.987	0.989	0.959	0.994	0.991

Note: the results better than baseline are underlined

Table 11: F-score for different text preprocessing strategies for SMS dataset

	SMS		
Method	NB	SVM	NN
baseline	0.831	0.990	0.992
100 words	0.972	0.981	0.983
200 words	0.974	0.985	0.987
500 words	0.633	0.978	0.989
1,000 words	0.651	0.976	0.991
unigrams	0.660	0.986	0.986
bigrams	0.662	0.990	0.992
binary weights	0.979	0.989	0.992
no stemming	0.831	0.990	0.992
no stopword removal	0.803	0.991	0.992

Note: the results better than baseline are underlined

Table 12: F-score for different text preprocessing strategies for social network datasets

	Hyves			Twitter		
Method	NB	SVM	NN	NB	SVM	NN
baseline	0.309	0.895	0.908	0.968	0.874	0.955
100 words	0.870	0.825	0.900	0.897	0.904	0.916
200 words	0.759	0.818	0.833	0.934	0.916	0.925
500 words	0.215	0.838	0.892	0.963	0.922	0.940
1,000 words	0.321	0.876	0.907	0.965	0.903	0.931
unigrams	0.780	0.706	0.917	0.962	0.891	0.852
bigrams	0.555	0.894	0.903	0.973	0.879	0.952
binary weights	0.397	0.898	0.918	0.967	0.889	0.944
no stemming	0.309	0.895	0.914	0.968	0.874	0.953
no stopword removal	0.309	0.895	0.911	0.968	0.859	0.959

Note: the results better than baseline are underlined

Table 13: F-score for different text preprocessing strategies for hotel review datasets

	Positive hotel reviews			Negative hotel reviews		
Method	NB	SVM	NN	NB	SVM	NN
baseline	0.864	0.881	0.899	0.863	0.866	0.890
100 words	0.755	0.789	0.780	0.764	0.766	0.753
200 words	0.814	0.812	0.795	0.780	0.790	0.811
500 words	0.833	0.819	0.817	0.818	0.814	0.865
1000 words	0.851	0.856	0.880	0.814	0.846	0.875
unigrams	0.836	0.868	0.774	0.790	0.853	0.879
bigrams	0.867	0.880	0.902	0.862	0.863	0.857
binary weights	0.826	0.843	0.815	0.863	0.862	0.888
no stemming	0.864	0.881	0.888	0.863	0.866	0.889
no stopword removal	0.864	0.881	0.897	0.849	0.869	0.900

Note: the results better than baseline are underlined

To further study the balance of the performance on both classes, fake and legitimate, AUC was calculated as shown in Tables 14-17. Increasing the number of features improved the performance in terms of AUC, except e-mail and social network datasets trained using the NB classifier. A similar effect can be observed for using only unigrams or bigrams. Again, using trigrams improved the performance for most datasets.

Table 14: AUC for different text preprocessing strategies for e-mail datasets

	Enron			SpamAssassin		
Method	NB	SVM	NN	NB	SVM	NN
baseline	0.781	0.978	0.998	0.953	0.994	0.999
100 words	0.975	0.943	0.978	0.943	0.963	0.992
200 words	0.977	0.952	0.990	0.951	0.970	0.996
500 words	0.937	0.959	0.997	0.959	0.982	0.999
1,000 words	0.864	0.968	0.998	0.960	0.989	0.993
unigrams	0.914	0.982	0.999	0.952	0.995	1.000
bigrams	0.844	0.979	0.999	0.971	0.994	1.000
binary weights	0.793	0.975	0.999	0.960	0.995	1.000
no stemming	0.781	0.978	0.998	0.963	0.994	0.999
no stopword removal	0.798	0.980	0.998	0.959	0.994	0.999

Note: the results better than baseline are underlined

The binary weighting scheme improved the classification performance for e-mail and social network datasets, while the *tf-idf* weighting scheme was more effective for SMS and hotel review datasets. The use of stemming increased AUC in almost all experiments. Removing stopwords was also beneficial, except the NB classifier.

Table 15: AUC for different text preprocessing strategies for SMS dataset

	SMS		
Method	NB	SVM	NN
baseline	0.937	0.955	0.994
100 words	0.940	0.907	0.973
200 words	0.944	0.932	0.983
500 words	0.940	0.931	0.989
1,000 words	0.940	0.933	0.992
unigrams	0.931	0.921	0.993
bigrams	0.934	0.954	0.995
binary weights	0.935	0.952	0.994
no stemming	0.937	0.955	0.994
no stopword removal	0.845	0.958	0.993

Note: the results better than baseline are underlined

Table 16: AUC for different text preprocessing strategies for social network datasets

	Hyves			Twitter		
Method	NB	SVM	NN	NB	SVM	NN
baseline	0.616	0.908	0.964	0.678	0.803	0.903
100 words	0.925	0.847	0.950	0.787	0.746	0.806
200 words	0.942	0.841	0.922	0.820	0.802	0.867
500 words	0.601	0.858	0.955	0.790	0.818	0.869
1,000 words	0.633	0.891	0.961	0.711	0.815	0.900
unigrams	0.940	0.767	0.964	0.787	0.807	0.870
bigrams	0.795	0.906	0.963	0.782	0.807	0.918
binary weights	0.665	0.911	0.965	0.779	0.817	0.903
no stemming	0.616	0.908	0.964	0.678	0.803	0.887
no stopword removal	0.616	0.908	0.964	0.709	0.790	0.832

Note: the results better than baseline are underlined

Table 17: AUC for different text preprocessing strategies for hotel review datasets

	Positive hotel reviews			Negative hotel reviews		
Method	NB	SVM	NN	NB	SVM	NN
baseline	0.886	0.880	0.960	0.899	0.865	0.957
100 words	0.839	0.790	0.862	0.828	0.765	0.822
200 words	0.878	0.810	0.848	0.856	0.790	0.887
500 words	0.898	0.819	0.900	0.873	0.814	0.938
1,000 words	0.892	0.855	0.951	0.862	0.845	0.946
unigrams	0.878	0.868	0.955	0.825	0.855	0.956
bigrams	0.892	0.880	0.961	0.887	0.863	0.933
binary weights	0.862	0.841	0.936	0.893	0.861	0.956
no stemming	0.886	0.880	0.959	0.899	0.865	0.955
no stopword removal	0.890	0.879	0.961	0.872	0.870	0.956

Note: the results better than baseline are underlined

To sum up, the results of the experiments above demonstrate the central importance of text preprocessing strategies in detecting spam / legitimate messages. The results indicate that common patterns can be observed, irrespective of both the used classifier and the classification domain. The number and length of the extracted word segments have major effect on the performance of the classifiers. Therefore, it is strongly recommended to use the sufficient number of word segments either in the form of bigrams or trigrams. In addition, the stemming and stop words removal techniques should be applied. The remaining technique, the non-binary weighting scheme may also slightly improve the results.

6. Deep Neural Network Model for Spam Filtering

Complex tasks require NNs with many hidden units to model them accurately (Murata et al., 1994). DNNs with many parameters are extremely powerful machine learning systems that contain multiple hidden layers to process complicated relationships between inputs and outputs (Schmidhuber, 2015). However, the large number of these relationships leads to sampling noise. As a result, complex adaptation to training data may lead to overfitting, preventing high accuracy on testing data. Overfitting can be effectively addressed through dropout regularization. In dropout, the units (hidden and visible) in a NN are temporarily removed from the network, including all their incoming and outgoing connections. In the fully connected layers of a feed-forward NN, dropout regularization randomly sets a given proportion (usually half) of activations to zero during training, thus potentially omitting hidden units that activate the same output.

Commonly used sigmoidal units reportedly suffer from the vanishing gradient problem, often accompanied by slow convergence of optimization to a poor local minimum (Maas et al., 2013). Rectified linear (ReL) units tackle this problem. When activated above 0, their partial derivative is 1. Moreover, ReL units saturate upon reaching 0, a characteristic that might be helpful in scenarios in which hidden activations are used as input features for the classifier. The ReL function can be defined as follows:

$$h_i = \max(w_i^T x, 0) = \begin{cases} w_i^T x & \text{if } w_i^T x > 0 \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

where h_i is the output of the activation function, $w_i^T x$ is the transpose of the weight vector of the i^{th} hidden unit and x is the input vector. The ReL function is therefore one-sided and does not enforce a sign symmetry or anti-symmetry. The main disadvantage of ReL is the fact that an NN using this function can easily produce sparse representation. In addition, such a NN has less intensive computation, exploiting the sparsity by avoiding the need to compute the exponential function in activations. The combination of dropout regularization and ReL units has shown promising synergistic effects (Jaitly and Hinton, 2011).

To find a suitable DFFNN structure, different numbers of hidden layers (from 1 to 3) and units in the hidden layers (from 10 to 200) were examined (see Figure 2 for an illustration). Training of the regularized DFFNN with ReL was performed using the mini-batch gradient descent algorithm, which updates the synapse weights θ for every mini-batch b of m training examples as follows:

$$\theta_{t+1} = \theta_t - \eta\nabla_\theta J(\theta_t d^{(i:i+m)} c^{(i:i+m)}), \tag{4}$$

where every mini-batch includes m training examples ($d^{(i)}$, $c^{(i)}$), i is the index of the training example within the minibatch, $c^{(i)}$ is the target class of the i-th training example, θ are the synapse weights of the DFFNN, $J(\theta_t)$ is an objective function to be minimized w.r.t. to the synapse weights θ_t, t represents time (iteration), and η denotes learning rate.

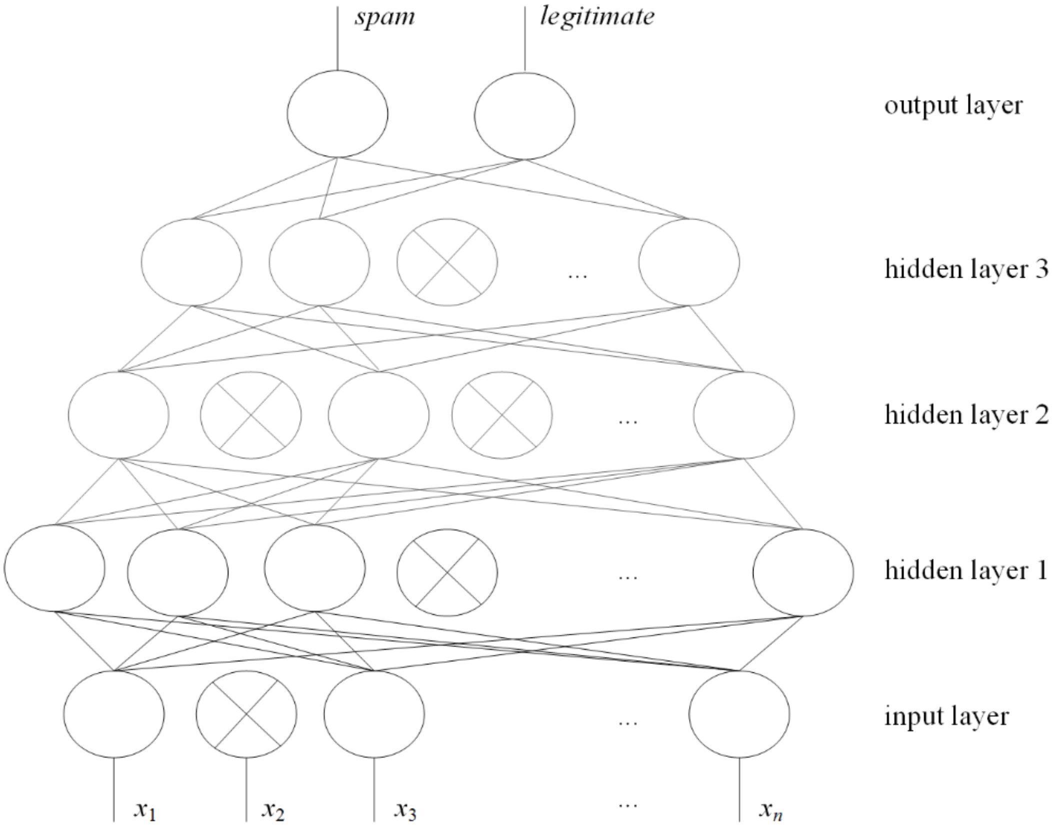

Figure 2: The structure of regularized DFFNN with ReL units for spam filtering (crossed neurons are dropped)

In the output layer, the following softmax function was used:

$$P(y_j) = \frac{\mathrm{e}^{\theta j}}{\sum_{k=1}^{K} \mathrm{e}^{\theta k}}, \tag{5}$$

where θ is the set of model parameters, and *j* and *k* denote the indexes of classes. Cross-entropy loss was used to represent objective function *J*.

The time complexity of the proposed DFFNN model is $O(n_b \times T \times (m \times n_1 + n_1 \times n_2 + n_2 \times n_3))$, where n_b is the number of mini-batches, *m* is the number of features, n_1 and n_2 are the numbers of neurons in the first and second hidden layer, respectively and n_3 is the number of neurons in the output layer.

This algorithm reduces the updates' variance, thus achieving a more stable convergence. Additionally, calculating the gradient w.r.t. a mini-batch makes this algorithm highly effective because it utilizes highly optimized matrix optimizations present in deep learning. The structure and parameters of the regularized DFFNN learning were found using a grid search procedure.

In order to further improve algorithms classification performance ensemble learning is applied with DFFNN as a base learner. The goal of ensemble learning algorithms is to combine the predictions of multiple base estimators constructed with the defined learning algorithm. This approach leads to better generalizability and robustness over single estimators. There are two main classes of ensemble learning algorithms, averaging and boosting. The fundamental concept of averaging is to construct several estimators independently from each other and calculate the average of their predictions. By reducing variance, the combined estimator is more accurate than single base estimator. By contrast, boosting builds the base estimators sequentially. Thus, several sequential weak models are combined to achieve a good ensemble. Here I use three conventional ensemble learning algorithms, namely Adaboost M1 (Freund and Schapire, 1996), Bagging (Breiman, 1996) and Random Subspace (Ho, 1998).

The Adaboost M1 algorithm was developed to produce predictions with high accuracy utilizing a number of weak base learners. The algorithm keeps building the learners until there are no errors in training data predictions or the limit numbers of models is exceeded. This is done by increasing the weights of incorrectly predicted data. Finally, the predictions from all the models are combined by using a weighted majority vote to obtain the final predictions. The algorithm is defined as follows:

```
Algorithm 1: Adaboost M1 with DFFNNs as base learners
Input: The set $D$ of training data ($x^i$; $y^i$), $i$=1,2, … ,$m$; the number $B$ of base DFFNNs
Output: Ensemble of base DFFNNs {$C_b$}
For $b$=1 to $B$ {
  Construct a base DFFNN $C_b$ on weighted training data $D^{\star}=(w_1 D^1{}_b, w_2 D^2{}_b, \ldots, w_m D^m{}_b)$;
  Calculate the probability estimates of the error
  $e_b=1/m\ \Sigma w_{ib}\times\xi^i{}_b$ ($\xi^i{}_b$=0 if $D^i$ classified correctly, $\xi^i{}_b$=1 otherwise);
  Set weight $c_b=0.5\times\log((1-\mathrm{err}_b)/\mathrm{err}_b)$;
  If $\mathrm{err}_b<0.5$, set  $w_{ib+1}=w_{ib}\times\exp(c_b\xi^i{}_b)$;
  Otherwise, set all weights $w_{ib}$=1 and restart the algorithm;
}
Combine base DFFNNs $C_b$, $b$=1,2,…,$B$ into an ensemble {$C_b$} by weighted majority voting;
```

The main idea behind Bagging is to construct multiple instances of black-box estimator on the random subsets of the original training data. To produce an aggregated prediction, separate predictions are then combined by using the voting procedure. Thus, the variance of base estimator is reduced by applying randomization during the process of building ensembles. The Bagging algorithm employed here can be defined as follows:

```
Algorithm 2: Bagging with DFFNNs as base learners
Input: The set $D$ of training data ($x^i$; $y^i$), $i$=1,2, … ,$m$; the number $B$ of base DFFNNs
Output: Ensemble of base DFFNNs {$C_b$}
For $b$=1 to $B$ {
  Create a bootstrapped replicate $D_b$ of the training data set $D$;
  Construct a base DFFNN $C_b$ on $D_b$;
}
Combine base DFFNNs $C_b$, $b$=1,2,…,$B$ into an ensemble {$C_b$} by simple majority voting;
```

Random Subspace (RSS) algorithm was proposed to handle the problem of trade-off between overfitting and achieving the highest accuracy. In fact, the RSS algorithm is similar to Bagging. The main difference is in the way they draw the random subsets of training data. In random subspace, these subsets are produced as the random subsets of the features. The RSS algorithm applied here for spam filtering can be defined as follows:

```
Algorithm 3: Random subspace with DFFNNs as base learners
Input: The set $D$ of training data ($x^i$; $y^i$), $i$=1,2, … ,$m$; the number $B$ of base DFFNNs
Output: Ensemble of base DFFNNs {$C_b$}
For $b$=1 to $B$ {
   Select an $r$-dimensional random subspace $D_b$ from the original training data set $D$;
   Construct a base DFFNN $C_b$ in $D_b$;
}
Combine base DFFNNs $C_b$, $b$=1,2,…,$B$ into an ensemble {$C_b$} by simple majority voting;
```

The time complexity of the proposed method can be obtained as follows. The time complexity of the ensemble-based DFFNN is $O(B \times n_{\mathrm{mb}} \times T \times (n \times n_1 + n_1 \times n_2 + n_2 \times n_3 + n_3 \times n_4))$, where B is the number of the base learners, n_{mb} is the number of mini-batches, T is the number of epochs, n is the number of features, n_1, n_2 and n_3 are the numbers of neurons in the first, second and third hidden layer, respectively, and n_4 is the number of neurons in the output layer.

Traditional machine learning algorithms use message content and other features to detect spam while not taking into consideration linguistic context of the words. In order to enhance the performance of spam detection, both bag-of-words and word context are taken into consideration in this work. More precisely, the proposed approach utilizes n-grams and the Skip-Gram word embedding method to build a vector model. Word2Vec (Mikolov et al., 2013; Le and Mikolov, 2014) is a popular method to produce word embeddings (vector space model) from a corpus of text data. As a result, high-dimensional feature representation is generated. To train the Skip-Gram model, I used the hierarchical softmax algorithm, a computationally effective version of the softmax algorithm. To further enhance the detection performance, I combined the generated word embeddings with bag-of-words in the second stage and train a DFFNN to classify spam/legitimate messages. Recall that DFFNN is used to capture complex features hidden in high-dimensional data representations (Barushka and Hajek, 2016; Barushka and Hajek 2018a, Barushka and Hajek, 2018b).

In the n-gram model, I used the BoW representation as defined in Eq. (1). In this model, text is represented as the bag of its words, disregarding grammar and even word order but keeping multiplicity. In BoW, string attributes are converted into a set of numeric attributes representing word occurrence information from the text contained in the strings. Note that only most relevant terms (attributes) were selected according to their weights w_{ij}. Top 2,000 terms were retained, including bigrams and trigrams as suggested by Li et al. (2017). To obtain word embeddings, the Skip-Gram model was employed. This is a language modelling and feature learning technique that maps words or phrases from the vocabulary to vectors of numerical values. Word embeddings are unsupervisedly learned word representation vectors whose relative similarities correlate with semantic similarity. The Skip-Gram model, one of the Word2Vec methods, includes the following steps (Mikolov et al., 2013; Le and Mikolov 2014):

- obtain a training dataset (sequences of words) $w_1; w_2; \ldots; w_T$;
- train the classifier and embedding function parameters;
- process each word w_t in the vocabulary by applying embedding function to generate digital representation for every word in the vocabulary in high-dimensional space;
- map every word in the vocabulary to digital representation of the word.

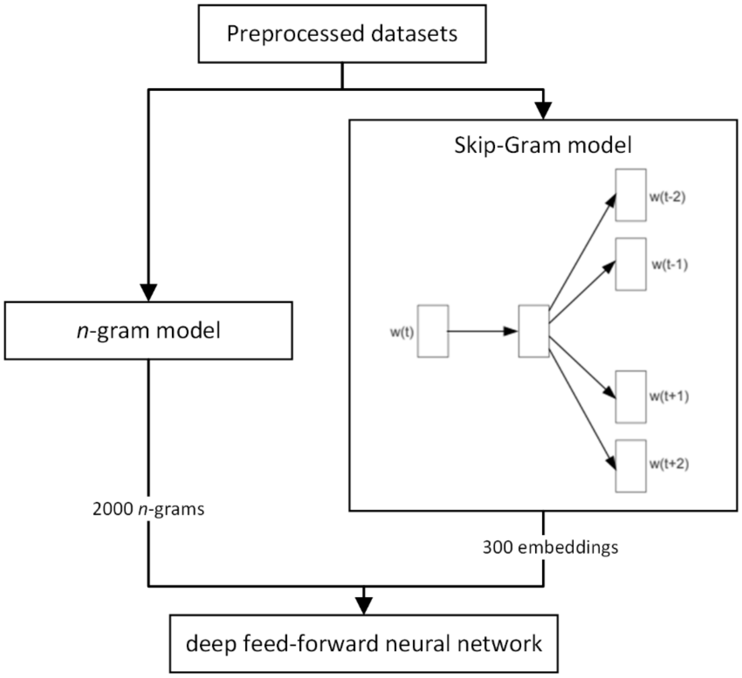

Figure 3: The proposed architecture of feature selection for spam filtering

The Skip-Gram model aims to find word representations that can be used to predict the context words in a sentence. The objective function of the skip-gram model is defined as follows:

$$E = \frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c}\log p\left(w_{t+j}\middle|w_t\right), \tag{6}$$

where w_1; w_2; . . .; w_T is a sequence of training words, c is the size of context, and $p(w_{t+1}|w_t)$ is defined using the hierarchical softmax (a binary tree representation of the output layer) as follows (Mikolov et al., 2013):

$$p(w|w_I) = \prod_{j=1}^{L(w)-1} \sigma(⟦n(w,j+1) = \mathrm{ch}(n(w,j))⟧ v'^{T}_{n(w,j)} v_{w_I}), \tag{7}$$

where w_I are input words, v_w and v'_w are the input and output vector representations of word w, respectively, $n(w, j)$ is the j-th node in the tree, $L(w)$ is the length of the path from root node to word w, ch(n) is a child node of n chosen arbitrarily, [x]=1 if x is true, otherwise [x]=-1, and $\sigma(x)$ is a sigmoidal function. Given the vocabulary size V, the computational complexity per training example per context word is $O(log(V))$, which is a substantial improvement over the original softmax ($O(V)$). The size of the word vectors (embeddings) was set to 300 and context size c = 5 (Mikolov et al., 2013) to generate a complex representation. The average values of the vector were used to represent each message. Thus, the input attributes (features) for the subsequent supervised learning included 2,000 *n*-grams and 300 embeddings.

To sum up, the proposed DFFNN model was represented by a multilayer perceptron NN with one to three hidden layers (Figure 4). DFFNNs can effectively process complex sparse representations of text documents just like spam and legitimate messages (Barushka and Hajek, 2018b). In the input layer of the proposed DFFNN model, two sets of features were extracted from the raw message text, namely (1) the top 2,000 unigrams, bigrams and trigrams according to their *tf.idf* weights, and (2) average 300 embeddings calculated for each message from the pre-trained embedding weight matrix (lookup table).

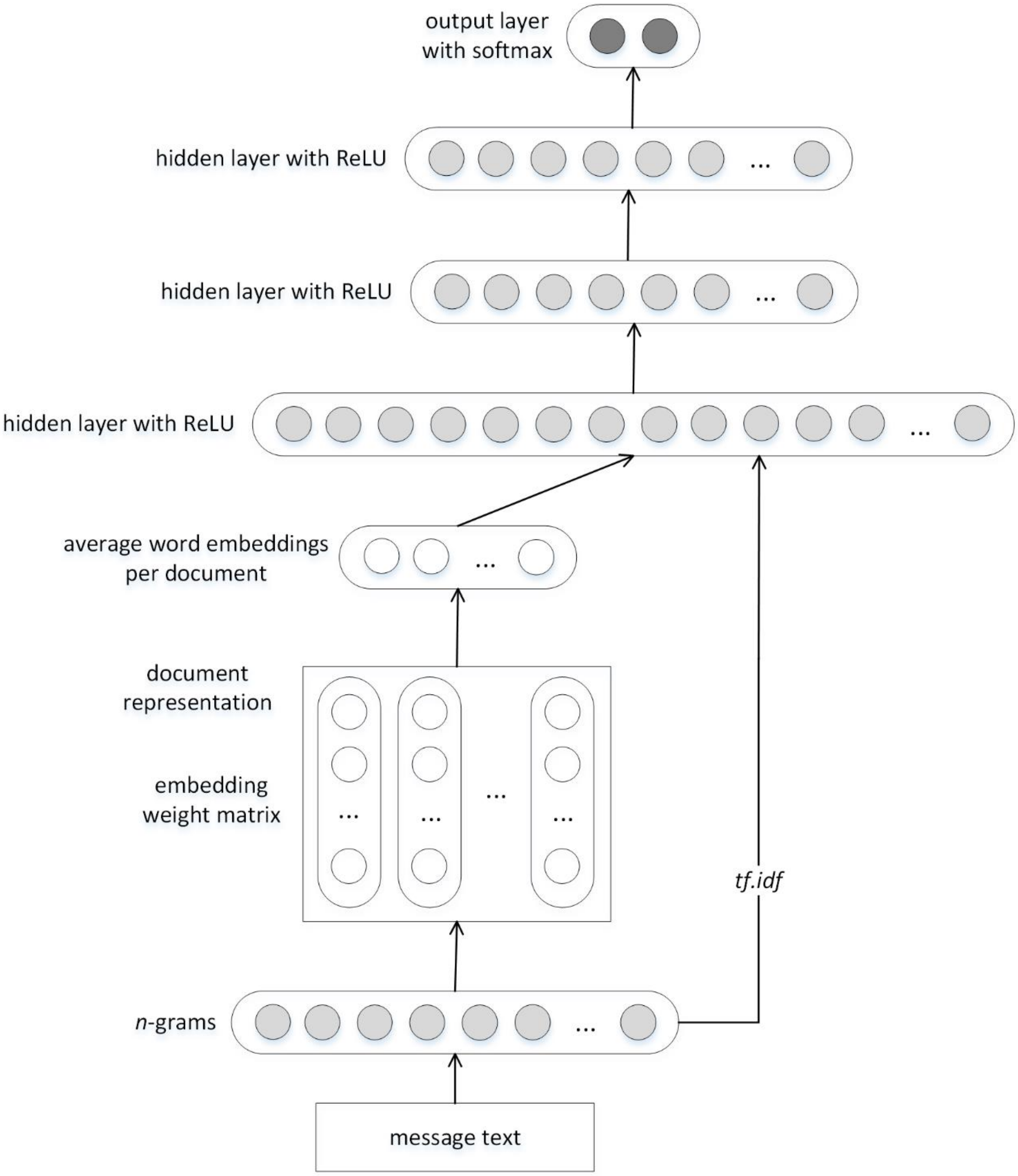

Figure 4: DFFNN model for spam filtering

7. Comparative Spam Filtering Models

To demonstrate the effectiveness of the proposed spam filtering model, the results are compared with recent approaches developed for spam classification, namely:

1) NB using factorial design analysis (Aragao et al., 2017),
2) SVM using factorial design analysis (Aragao et al., 2017),
3) Incremental Learning with C4.5 (Sheu et al., 2017),
4) RF (Khorshidpour et al., 2017),
5) Voting (Najadat et al., 2016),
6) CNN (Ren and Ji, 2017).

These comparative methods were used as they represent the state-of-the-art machine learning approaches to spam filtering with supervised learning. These methods are briefly described below. In addition, several traditional machine learning methods are used, such as *k*-NN, Bagging and AdaBoost M1 to include all types of machine learning methods presented in previous review studies (Guzella and Caminhas, 2009; Pérez-Díaz et al., 2012).

7.1 Factorial Design Analysis using NB and SVM

The NB classifier, a probability-based approach, has become a popular method for spam filtering due to its simplicity (Metsis et al., 2006). It uses information learned from training data to compute the posterior probability that a message is spam or legitimate given the words that appear in the message. However, NB relies on the assumption that feature values are conditionally independent given the class, an assumption which often does not hold in text classification tasks. Overall, NB learning is relatively easy to implement and accommodates discrete features reasonably well.

SVMs are reportedly effective classifiers for spam filtering due to their ability to handle high-dimensional data (Lai, 2007). They find the optimal separating hyperplane that provides the maximum margin between two classes. A subset of the training data (the so-called support vectors) are used to define the decision boundaries. SMO is a frequently used technique to find the parameters of the separating hyperplane. This algorithm decomposes the overall quadratic programming problem into sub-problems, using Osuna’s theorem to ensure convergence. In cases of non-linear classification, kernel functions are used to map the problem from its original feature space onto a new feature space where linear separability is ensured.

In the spam filter proposed by Aragao et al., (2017), factorial design analysis (FDA) is used to obtain the optimal filter setup. Specifically, FDA finds the best combination of three text pre-processing parameters for SVM and NB classifiers. The parameters are represented by stop-words removal (yes/no), lemmatization (yes/no), and the number of features (128/1024), leading to 2^3 factorial design matrix. In Aragao et al., (2017), SVM-based spam filter performed better without stop-words removal and lemmatization, whereas these linguistic techniques were effective for the NB classifier. For both spam filters, performance increased with a high level of features.

7.2 Incremental Learning with C4.5

The J48 training algorithm is a popular version of the well-known C4.5 decision tree (Quinlan 1996). J48 generates a decision tree model with varying classification rates based on cross-validation. Using fewer features to create the model may benefit performance efficiency by minimizing the number of branches on the tree which must be calculated. In this dissertation thesis, I use an incremental learning mechanism using C4.5 (IL C4.5) proposed to better adapt to the dynamic environment (Sheu et al., 2017). In this algorithm, a critical attribute is selected based on the maximum value of Gain Ratio, and the base of association rules is formed using the paths from root nodes to leaf nodes.

7.3 Random Forest

Recently, it was shown that RF are effective classifiers in spam filtering owing to its non-differentiable decision boundary (Khorshidpour et al., 2017). RF (Breiman, 2001) combines tree predictors in such a way that each single tree depends on the values of a random vector sampled independently from the others, and all trees in the forest have the same distribution. Once the number of trees in the forest grows large enough, the generalization error for the forest converges to a limit. The generalization error depends on two factors: the strengths of individual trees and the correlations between them. Using a random selection of features to split each node yields error rates that compare favorably to AdaBoost, but that are more robust with respect to noise, thus improving the performance of a spam filter (Koprinska et al., 2007).

7.4 Voting

Voting is an ensemble method, combining the decisions of several base learners. Here I use the combination of NB, SVM, and Stochastic Gradient Descent classifiers proposed for SMS spam filtering in Najadat et al. (2016). This approach employs majority voting, and it was reported to be more effective in spam filtering than the above-mentioned classifiers trained individually. This was attributed to computational effectivity, fast convergence, and resiliency to overfitting (Najadat et al., 2016).

7.5 Convolutional Neural Network

Convolutional neural network A CNN is a variant of DFFNN, utilizing layers with convolving filters that are applied to the local features of adjacent layers (LeCun et al., 1998). The filters in any given layer form a feature map and share the same parametrization. Each hidden layer comprises multiple feature maps, obtaining a complex data representation. To capture the most important feature for each feature map, a max-pooling operation is applied over that map. Although originally developed for the computer vision domain, CNNs have recently shown effectiveness in text-categorization tasks (Kim, 2014). Despite this interest in their use in general text categorization, to the best of my knowledge CNNs have only been applied to review spam detection (Ren and Ji, 2017). In agreement with this previous study, a CNN model is used by employing the pre-trained CBOW model with 300 word embeddings.

7.6 Other Machine Learning Methods

Another simple machine learning method used for spam filtering is the ***k-NN classifier***. Considered an example-based classifier, training data are used for comparison rather than to explicitly represent class. There is basically no training phase. A new message is classified based on the *k* most-similar messages (typically using Euclidean distance). Moreover, finding the nearest neighbor(s) can be accelerated using indexing. However, other machine learning methods usually outperform this algorithm in spam filtering (Zhang et al., 2014).

AdaBoost, the first practical boosting algorithm, remains one of the most widely used and studied such algorithms, with applications in numerous fields (Freund et al., 1999). Regarding machine learning, boosting means obtaining a prediction with high accuracy by combining a set of relatively weak and inaccurate rules. A first model is built from the training data, and then a second model is created to correct the errors of the first model. Iterative models are created until either the training set is predicted without errors or the maximum number of models is reached. In this way, highly accurate spam filters can be developed, as shown by the comparative study performed in Zhang et al. (2014).

The main idea behind ***Bagging*** is to construct multiple instances of black-box estimator on the random subsets of the original training data. To produce an aggregated prediction, separate predictions are then combined by using the voting procedure. Thus, the variance of base estimator is reduced by applying randomization during the process of building ensembles.

In Table 18, the compared methods are presented regarding their capacity to deal with high-dimensional and sparse datasets. Recall that spam / legitimate messages are generally short texts, thus corresponding to sparse datasets. Moreover, to represent the linguistic features of the texts, high-dimensional feature vectors must be generated. Therefore, these two data characteristics are crucial for effective spam filtering machine learning methods.

Table 18: Summary of compared methods regarding data characteristics

Method	high dimensionality of data	data sparsity
FDA+NB	+	+
FDA+SVM	+	+
IL+C4.5	–	+
Voting	+	+
RF	+	+
CNN	+	+
k-NN	–	–
AdaBoost	–	+
Bagging	–	+
FFNN	+	–
DFFNN	+	+
Ensemble learning with DFFNNs as base learners	+	+

Legend: + for strength and – for weakness

8. Experimental Settings

8.1 Hardware and Software Specification

All the experiments were performed on a hardware server with 2 CPU sockets. In each CPU socket, AMD Opteron Processor 6180 SE[8] was installed. The CPUs were running on 2.50 GHz frequency and had 12 cores (threads). Each CPU had 1.5 MB L1 cache, 6 MB L2 cache and 12 MB L3 cache. There were 16 DDR3 RAM memory cards installed and each card had a capacity of 16 GB.

The server was running the 64-bit version of Windows 10 (Educational version) operating system. The experiments were run in Weka 3.8.3 x64 program environment. Specifically, text preprocessing was conducted using the StringToWordVector library, the embeddings were trained using the Dl4jStringToWord2Vec library, and all the DNNs (DFFNNs and CNNs) were implemented in the Deeplearning4j Java library. This program environment required Java virtual machine and Java version 8 Update 181 (build 1.8.0_181-b13) installed on the server.

8.2 Data Preprocessing

To select the suitable data preprocessing strategy for the BoW features, accuracies from Tables 6-9 were analyzed. If at least two classifiers performed better than the baseline setting, the data preprocessing strategy was modified. As a result, the following data preprocessing techniques were used for the datasets:

- Enron – 2,000 unigrams with binary weights, stemming no stopword removal;
- SpamAssassin – 2,000 unigrams with *tf.idf* weights, stemming, stopword removal;
- SMS – 2,000 unigrams+bigrams with *tf.idf* weights, stemming, no stopword removal;
- Hyves – 2,000 unigrams+bigrams+trigrams with binary weights, stemming, stopword removal;
- Twitter – 2,000 unigrams+bigrams with binary weights, stemming, no stopword removal;
- Positive hotel reviews – 2,000 unigrams+bigrams with *tf.idf* weights, stemming, stopword removal;

[8] https://www.amd.com/en/products/cpu/6180-se

- Negative hotel reviews – 2,000 unigrams+bigrams+trigrams with *tf.idf* weights, stemming, no stopword removal.

For example, features with the highest values of information gain are presented in Table 19 and Table 20. Note that *n*-grams are not presented for the Hyves datasets because in each word was assigned an anonymized id in the source data files.

To perform the training process of the Skip-Gram model, the corresponding datasets were used to obtain word embeddings. However, for the Hyves and hotel review datasets, the size of the data was insufficient. Therefore, a large corpus of ~84 million Amazon reviews[9] was used to pre-train the word embeddings for the hotel review datasets. The problem with the Hyves dataset was that the words were represented only by their id in the original dataset, therefore the Skip-Gram model was only trained on this original dataset of small size. As a result, one can expect a worse word representation for this dataset.

Table 19: Top 10 features from the *n*-gram model for e-mail and SMS datasets in terms of information gain

Enron		SpamAssassin		SMS	
feature	IG	feature	IG	feature	IG
„enron“	0.170	„list-id“	0.591	„call“	0.057
„2000“	0.136	„mailman-version“	0.587	„free“	0.048
„cc“	0.131	„beenthere“	0.580	„www“	0.042
„hpl“	0.121	„errors-to“	0.572	„mobile“	0.039
„daren“	0.112	„precedence“	0.549	„claim“	0.036
„http“	0.101	„bulk“	0.502	„prize“	0.035
„gas“	0.099	„IMAP“	0.480	„txt“	0.035
„forwarded“	0.095	„localhost“	0.480	„&”	0.033
„-forwarded“	0.095	„fetchmail-5“	0.480	„stop“	0.029
„pm”	0.093	„received“	0.407	„won“	0.025

[9] http://jmcauley.ucsd.edu/data/amazon/

Table 20: Top 10 features from the *n*-gram model for social network and hotel review datasets in terms of information gain

Twitter		Negative hotel review		Positive hotel reviews	
feature	IG	feature	IG	feature	IG
„more for“	0.060	„chicago“	0.123	„chicago“	0.090
„invisible >&nbsp“	0.036	„at the“	0.046	„location“	0.062
„data-expanded-url= http“	0.033	„luxury“	0.044	„floor“	0.052
„http“	0.033	„location“	0.043	„bathroom“	0.044
„title= http“	0.033	„-“	0.038	„on the“	0.041
„class= js-display-url“	0.033	„when i“	0.036	„small“	0.037
„invisible ></span>“	0.032	„chicago hotel“	0.034	„reviews“	0.037
„get weather“	0.026	„smell“	0.033	„luxury“	0.037
„weather updates“	0.026	„my room“	0.033	„2“	0.037
„updates from”	0.026	„recently”	0.030	„priceline“	0.035

8.3 Data Partitioning

In order to benchmark the algorithm performance, datasets are usually split into training and testing subsets. Once model is trained on the training dataset, the testing dataset is used to evaluate it. However, performing this process only once and randomly has a serious limitation and can lead to sample selection bias (Kohavi, 1995). To tackle this problem, *K*-fold cross validation was introduced. Following this approach, the dataset is randomly split into *K* equally sized parts. After that, the model is trained *K* times. For each training cycle, a single partition is selected which has not been selected in the previous cycles. The selected part is used for testing and the rest of the dataset are used for training. Therefore, each model will be trained and tested on a unique training dataset. Once all *K* cycles are run, the results are summed up. Studies suggest that setting 10 as the *K* value provides reliable results preventing both excessively high bias and variance (Kohavi, 1995).

8.4 Settings of Machine Learning Methods

For the FDA, the parameters were represented by stop-words removal (yes/no), lemmatization (yes/no), and the number of features (200/1000). In agreement with Aragao et al. (2016), SVM and NB were used as classifiers in the FDA framework. The LibLINEAR implementation of the L2-regularized L2-loss SVM was used for the experiments. In the experiments, SVMs were tested with a polynomial kernel function and complexity parameter $C = \{2^0, 2^1, 2^2, \dots , 2^8\}$.

To train the Incremental Learning with C4.5 (IL+C4.5) spam filter, I used the J48 implementation of the C4.5 algorithm with confidence factor = 0.25 and minimum number of instances per leaf = 2. Following the selection of base learners used in Najadat et al. (2016), NB, SVM and Stochastic Gradient Descent algorithms were used in Voting. The setting of the SVM was the same as for the FDA, while Hinge loss function was used in the Stochastic Gradient Descent algorithm.

RF worked with 100 random trees. The *k*-NN classifier with the Euclidean distance function and number of neighbors set to *k* = 3. The AdaBoost M1 version was trained with Decision Stump as base learners and the number of iterations was 10. Bagging was trained with REPTree as the base learner.

As with the FFDNN, the CNN was trained using a mini-batch gradient descent algorithm with patch size 5×5 and max pool size 2×2, each with number of feature maps = {10, 20, 50, 100, 200}; learning rate = 0.05; size of each mini-batch used in computing gradients *b* = 100; input layer dropout rate = 0.2; hidden layer dropout rate = 0.5; and number of iterations = 1,000.

8.5 Evaluation Measures

While evaluating the experimental results, the following evaluation measures were taken into consideration: Accuracy rate Acc, FPR, FNR, AUC, F-score, misclassification cost (MC) and computational time (training and testing time).

Accuracy rate (Acc) is the percentage of messages which were predicted correctly. Accuracy rate can be calculated using the formula below:

$$\text{Acc} = \frac{TP+TN}{FP+FN+TP+TN}, \tag{8}$$

where *TP* is the number of true positives, *TN* is the number of true negatives, *FP* is the number of false positives and *FN* is the number of false negatives.

FNR is the percentage of legitimate messages incorrectly predicted as spam. FNR can be calculated as follows:

$$FNR = \frac{FN}{TP+FN}. \tag{9}$$

FPR represents the percentage of spam messages incorrectly predicted as legitimate. It can be calculated using the following formula:

$$FPR = \frac{FP}{FP+TN}. \tag{10}$$

The F-score combines precision and recall, where precision is a fraction of messages correctly classified as spam out of all the messages the algorithm classifies as spam, whereas recall is the fraction of messages correctly classified as spam out of all the spam messages. The F-score is calculated as follows:

$$F - score = 2 \times \frac{precision \times recal}{precision + recal}. \tag{11}$$

ROC is a graphical representation which shows the performance of a classification model at all classification thresholds. The ROC curve is created by plotting TPR against FPR at various threshold settings (Figure 5). AUC represents the two-dimensional area underneath the entire ROC curve. In other words, AUC represents the probability that the classifier ranks a randomly chosen legitimate message higher than a randomly chosen spam message.

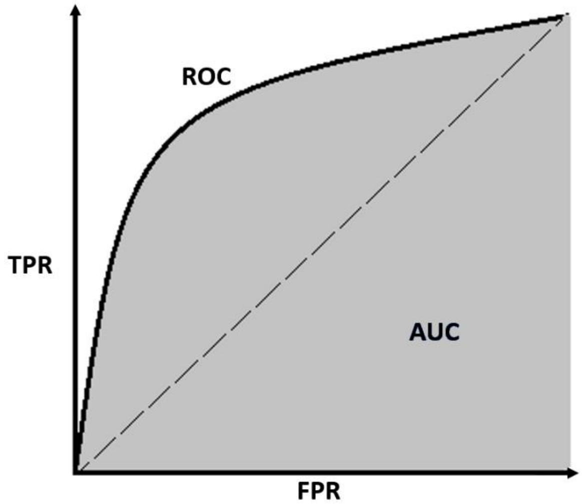

Figure 5: ROC curve and AUC[10]

[10] https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

In the literature on credit risk modelling, AUC was reported to be a suitable performance measure, mainly because it is robust against imbalanced data:

$$AUC = \int_0^1 TPR(T) * \frac{d}{dT} FPR(T) dT, \tag{12}$$

where T is any cut-off point, $0 < T < 1$. On the one hand, the wrong prediction of a message that is spam (type II error) leads to the loss of time because the user needs to read the message, delete it and report a spam message (or spamming profile), respectively. On the other hand, predicting a spam message when it would be legitimate (type I error) may result in its automatic filtering and ignoring by the user or eventually in its automatic deletion. This case is considered more serious than the former one because we want avoid labelling legitimate message as spam (Zhang et al., 2014). Several spam filtering studies have combined those two errors into a misclassification cost (MC), which is considered a crucial criterion in the evaluation of spam filtering effectiveness (Jia and Shang, 2014). However, this measure has rarely been utilized as the evaluation criterion in spam filtering models (Zhang et al., 2014).

Table 21 shows the confusion matrix used to calculate MC, which combines type I and type II errors as follows:

$$MC_{\lambda} = \frac{1}{1+\lambda} \times FPR + \frac{\lambda}{1+\lambda} \times FNR\ , \tag{13}$$

where λ is a misclassification cost ratio comparing the degree of seriousness of type I error compared to type II error.

Table 21: Confusion matrix for spam filtering

Prediction/Actual	Negative	Positive
Negative (spam)	TN	FN (type I error)
Positive (legitimate)	FP (type II error)	TP

Legend: TP, FP, FN and TN are the numbers of messages classified as true positive, false positive, false negative and true negative.

Machine learning algorithms tend to be computing resource intensive, especially in terms of CPU time. While model testing takes insignificant amount of CPU time, model training may take considerable amount of time. Training (testing) time is evaluated using the amount of time spent on learning (testing) in milliseconds.

9. Experimental Results

In this chapter, I present the results of experiments performed to empirically evaluate the effectiveness of the proposed spam filtering models on the seven benchmark datasets. Hereinafter, the averages and standard deviations of the stratified 10-fold cross-validation are presented. To compare the results of the proposed models with existing approaches presented above, the results are presented for each evaluation measure.

9.1 Performance of Spam Filtering Methods in terms of Accuracy

As presented in Table 22, the proposed methods with ensemble learning showed the best performance for six out of seven datasets. For the remaining dataset Twitter, the proposed DFFNN model performed best. The results demonstrate that ensemble learning based on Bagging and RSS produce the highest accuracies. Notably, the ensemble methods with DFFNN as base learners performed substantially better than those with DTs as based learners. Furthermore, DFFNN also performed better than the CNN model for most datasets.

Standard deviations of accuracy for the proposed spam filtering models were also lower than for the compared models, indicating a good stability of the proposed models. Most importantly, there is a strong consistency in the performance of the proposed model across all datasets, which suggests that the proposed models produce high accuracies for e-mail, SMS, social network and hotel review datasets. The highest accuracy was achieved for the e-mail datasets, while the worst performance was obtained for the Twitter and hotel review datasets. This result confirms that it is more difficult to identify spam messages in social networks and online reviews. The results also show that the proposed models perform well for both balanced and imbalanced datasets.

Regarding the compared methods, FDA+NB performed well only for smaller and balanced datasets, namely Hyves and hotel reviews. FDA+SVM, Voting and Bagging models also performed relatively well for the e-mail and SMS datasets. By contrast, the *k*-NN and Adaboost M1 models performed relatively poorly.

Table 22: Performance of spam filtering methods in terms of accuracy [%]

Method	Enron	SpamAssasin	SMS	
FDA+NB	86.66±1.71	94.67±1.09	95.77±0.98	
FDA+SVM	96.83±0.99	98.79±0.68	97.37±0.51	
IL+C4.5	94.02±1.11	96.35±1.20	96.11±0.62	
RF	96.04±1.35	97.36±0.94	97.57±0.74	
Voting	97.20±0.61	89.04±3.20	98.04±0.75	
CNN	94.08±2.50	97.21±1.05	93.80±4.97	
k-NN	91.72±1.39	96.57±1.04	92.94±0.69	
AdaBoostM1	78.73±1.17	94.75±1.09	88.66±0.72	
Bagging	95.38±0.79	96.78±1.02	96.73±0.70	
DFFNN	97.83±0.53	99.00±0.79	98.55±0.51	
AdaBoostM1+DFFNN	98.65±0.58	99.03±0.42	98.32±0.40	
Bagging+DFFNN	98.88±0.46	98.96±0.57	98.70±0.55	
RSS+DFNNN	99.05±0.37	99.14±0.48	98.50±0.67	
Method	Hyves	Twitter	Positive hotel reviews	Negative hotel reviews
FDA+NB	88.55±3.37	78.81±0.59	86.13±4.58	84.63±3.59
FDA+SVM	86.97±3.35	85.21±4.38	82.00±4.57	84.00±5.23
IL+C4.5	87.70±3.52	90.18±0.85	71.63±4.60	72.00±5.41
RF	89.28±2.14	86.78±0.92	73.88±5.12	70.63±5.84
Voting	90.38±2.11	84.71±1.89	84.63±3.91	86.50±4.99
CNN	91.96±2.32	80.39±4.68	79.75±4.99	76.75±3.34
k-NN	89.52±3.27	88.14±1.11	65.13±5.05	69.50±4.68
AdaBoostM1	89.04±3.20	84.39±0.32	67.13±5.87	73.63±4.10
Bagging	90.38±2.04	89.23±0.85	76.63±6.75	75.63±4.14
DFFNN	87.82±1.92	90.32±0.69	86.63±5.11	88.75±4.60
AdaBoostM1+DFFNN	91.47±2.38	86.65±1.77	85.50±5.93	86.13±3.36
Bagging+DFFNN	92.45±1.62	89.51±1.03	87.63±4.80	90.38±3.12
RSS+DFNNN	92.32±1.74	89.67±0.88	87.63±5.22	89.50±2.71

9.2 Performance of Spam Filtering Methods in terms of FNR and FPR

Table 23 and Table 24 show the performance of the compared models in terms of FNR and FPR, respectively. Recall that FNR (type I error) is considered more serious than FPR (type II error) because we want avoid to labelling legitimate message as spam.

Regarding FNR, the proposed algorithms showed the best performance for five out of the seven datasets. FDA+NB performed slightly better for the Enron and Hyves datasets. However, Table 24 shows that FDA+NB did not perform well for both classes on these two datasets. Moreover, FDA+NB also showed inconsistent results in terms of FNR because it had the worst score for the SpamAssasin dataset and below average performance for the rest of the datasets. The results show that the proposed models perform well for all datasets regardless the spam classification domain.

DFFNNs based on ensemble algorithms showed lower FNR then their DT counterparts for all the datasets. Moreover, the ensemble approaches using DFFNN outperformed the single DFFNN model for all the datasets. Voting and FDA+SVM also demonstrated consistent performance across the spam classification domain, while AdaBoost M1 performed poorly for the imbalanced datasets, including SMS, Enron and Twitter.

Table 23: Performance of spam filtering methods in terms of FNR

Method	Enron	SpamAssasin	SMS	
FDA+NB	0.001±0.002	0.082±0.021	0.124±0.038	
FDA+SVM	0.049±0.025	0.014±0.009	0.119±0.064	
IL+C4.5	0.071±0.020	0.039±0.015	0.239±0.045	
RF	0.057±0.017	0.034±0.022	0.158±0.055	
Voting	0.025±0.012	0.112±0.030	0.095±0.049	
CNN	0.028±0.010	0.018±0.017	0.084±0.038	
k-NN	0.050±0.027	0.051±0.018	0.528±0.052	
AdaBoostM1	0.685±0.025	0.059±0.019	0.820±0.046	
Bagging	0.051±0.015	0.037±0.015	0.187±0.048	
DFFNN	0.019±0.011	0.013±0.012	0.095±0.037	
AdaBoostM1+DFFNN	0.021±0.013	0.012±0.010	0.083±0.038	
Bagging+DFFNN	0.010±0.006	0.016±0.010	0.091±0.041	
RSS+DFNNN	0.009±0.007	0.014±0.009	0.099±0.051	
Method	Hyves	Twitter	Positive hotel reviews	Negative hotel reviews
FDA+NB	0.011±0.011	0.325±0.013	0.120±0.031	0.125±0.055
FDA+SVM	0.146±0.053	0.213±0.025	0.172±0.051	0.165±0.058
IL+C4.5	0.140±0.037	0.258±0.018	0.275±0.075	0.273±0.065
RF	0.099±0.044	0.217±0.019	0.245±0.070	0.267±0.069
Voting	0.112±0.030	0.210±0.023	0.150±0.057	0.117±0.046
CNN	0.086±0.044	0.500±0.527	0.207±0.290	0.185±0.067
k-NN	0.184±0.048	0.263±0.018	0.147±0.056	0.310±0.093
AdaBoostM1	0.167±0.041	0.383±0.008	0.268±0.084	0.258±0.083
Bagging	0.144±0.039	0.244±0.025	0.253±0.086	0.238±0.067
DFFNN	0.103±0.046	0.254±0.019	0.135±0.058	0.120±0.069
AdaBoostM1+DFFNN	0.097±0.037	0.210±0.021	0.140±0.075	0.142±0.065
Bagging+DFFNN	0.094±0.039	0.247±0.019	0.112±0.050	0.102±0.058
RSS+DFNNN	0.088±0.043	0.258±0.021	0.107±0.050	0.105±0.047

Concerning FPR, the proposed spam filtering models performed very well for all the datasets, being the best for the e-mail and hotel review datasets. In addition, they ranked among the best also for the SMS and social network datasets (Table 24).

The *k*-NN classifier performed best for two datasets, SMS and Hyves, indicating that this method classified most messages as legitimate. Indeed, the results for FNR above confirm this implication. This is due to the strong imbalance of these datasets. By contrast, DFFNN performed best for the Twitter dataset, while showing a good performance also in terms of FNR for this dataset. Overall, we can see that the proposed models based on DFFNNs performed well for both classes, this is in terms of both FNR and FPR. Only FNR for the Twitter dataset was greater than 0.200, indicating a good balance between type I and type II errors.

Table 24: Performance of spam filtering methods in terms of FPR

Method	Enron	SpamAssasin	SMS	Hyves	Twitter	Positive hotel reviews	Negative hotel reviews
FDA+NB	0.187±0.025	0.016±0.013	0.030±0.008	0.107±0.048	0.203±0.006	0.158±0.068	0.130±0.047
FDA+SVM	0.025±0.009	0.011±0.007	0.008±0.005	0.104±0.053	0.135±0.077	0.182±0.084	0.155±0.074
IL+C4.5	0.055±0.014	0.034±0.020	0.006±0.003	0.096±0.065	0.086±0.010	0.278±0.084	0.287±0.101
RF	0.030±0.009	0.019±0.012	0.004±0.002	0.113±0.061	0.125±0.010	0.278±0.086	0.280±0.090
Voting	0.029±0.006	0.107±0.069	0.008±0.005	0.107±0.069	0.149±0.022	0.158±0.068	0.152±0.088
CNN	0.019±0.007	0.009±0.007	0.003±0.002	0.088±0.050	0.300±0.483	0.142±0.050	0.107±0.054
k-NN	0.083±0.023	0.016±0.007	0.000±0.000	0.023±0.026	0.107±0.013	0.380±0.134	0.240±0.043
AdaBoostM1	0.020±0.009	0.035±0.012	0.005±0.003	0.028±0.027	0.139±0.004	0.357±0.126	0.270±0.050
Bagging	0.045±0.013	0.028±0.014	0.008±0.005	0.034±0.029	0.097±0.009	0.215±0.088	0.242±0.069
DFFNN	0.021±0.010	0.006±0.007	0.001±0.001	0.042±0.036	0.085±0.008	0.133±0.072	0.105±0.062
AdaBoostM1+DFFNN	0.011±0.005	0.007±0.008	0.007±0.003	0.071±0.043	0.128±0.019	0.150±0.068	0.135±0.052
Bagging+DFFNN	0.012±0.005	0.004±0.008	0.001±0.001	0.051±0.035	0.095±0.011	0.135±0.077	0.090±0.038
RSS+DFNNN	0.010±0.004	0.003±0.006	0.002±0.002	0.062±0.033	0.092±0.010	0.140±0.077	0.105±0.026

In fact, all methods performed well for the e-mail, SMS and Hyves datasets in terms of FPR, suggesting that spam messages can be easily identified using any of these machine learning methods. This was more difficult for Twitter and hotel reviews. As expected, finding spam in hotel reviews was the most demanding task because the authors of fake reviews indent to produce reviews as similar as possible to the legitimate ones. Obviously, DNNs achieved the highest accuracy on the spam class for the hotel review datasets.

9.3 Performance of Spam Filtering Methods in terms of AUC and F-score

Table 25 shows the performance of the spam filtering methods in terms of AUC, this is the performance measure that is, unlike accuracy, robust to class imbalance.

Table 25: Performance of spam filtering methods in terms of AUC

Method	Enron	SpamAssasin	SMS
FDA+NB	0.975±0.007	0.967±0.007	0.974±0.007
FDA+SVM	0.963±0.014	0.988±0.007	0.933±0.018
IL+C4.5	0.968±0.013	0.974±0.010	0.911±0.021
RF	0.989±0.003	0.991±0.005	0.977±0.017
Voting	0.995±0.002	0.947±0.015	0.983±0.010
CNN	0.997±0.001	0.999±0.001	0.982±0.015
k-NN	0.973±0.007	0.988±0.005	0.929±0.023
AdaBoostM1	0.897±0.010	0.989±0.005	0.798±0.033
Bagging	0.990±0.003	0.996±0.003	0.969±0.012
DFFNN	0.997±0.001	0.999±0.000	0.988±0.009
AdaBoostM1+DFFNN	0.998±0.002	0.997±0.002	0.983±0.012
Bagging+DFFNN	0.999±0.001	1.000±0.000	0.993±0.006
RSS+DFNNN	0.999±0.000	1.000±0.000	0.993±0.006

Method	Hyves	Twitter	Positive hotel reviews	Negative hotel reviews
FDA+NB	0.934±0.024	0.811±0.008	0.942±0.026	0.917±0.025
FDA+SVM	0.873±0.033	0.801±0.007	0.820±0.046	0.840±0.052
IL+C4.5	0.879±0.035	0.864±0.007	0.730±0.067	0.736±0.037
RF	0.944±0.021	0.898±0.006	0.808±0.045	0.783±0.053
Voting	0.947±0.015	0.872±0.005	0.934±0.026	0.935±0.027
CNN	0.944±0.017	0.630±0.107	0.925±0.034	0.909±0.033
k-NN	0.927±0.025	0.877±0.010	0.707±0.074	0.766±0.036
AdaBoostM1	0.906±0.019	0.751±0.008	0.756±0.057	0.817±0.046
Bagging	0.943±0.018	0.897±0.007	0.848±0.052	0.825±0.046
DFFNN	0.957±0.021	0.901±0.007	0.942±0.022	0.956±0.023
AdaBoostM1+DFFNN	0.950±0.017	0.904±0.007	0.919±0.048	0.935±0.023
Bagging+DFFNN	0.956±0.017	0.907±0.007	0.945±0.025	0.960±0.018
RSS+DFNNN	0.958±0.017	0.905±0.007	0.945±0.027	0.959±0.016

Notably, the proposed models showed the best performance for all the seven datasets. Bagging and RSS trained with DFFNN as base learner performed particularly well for all spam domains. CNN and Voting also performed well, whereas the remaining methods provided inconsistent performance in terms of both classes. The results also demonstrate that ensemble learning with DFFNN improves the overall performance compared with those using the DTs as base learners. The results of the proposed models are consistent across all datasets and the performance is solid for both balanced and imbalanced datasets and different classification domains.

Regarding F-score, the proposed spam filtering models also performed best except the Twitter dataset, suggesting that the overall performance is solid on the spam class in terms of both precision and recall (Table 26). In other words, the proposed filters not only detect the spam messages with a high accuracy but they also do not classify too much legitimate messages into the spam class. This indicates that the proposed provides a balance between spam precision and recall.

Bagging with DFFNN as base learners performed particularly well in all spam domains. Again, the worst performance can be observed for the hotel review datasets, confirming the difficult identification of fake reviews. By contrast, the *k*-NN method performed worst, indicating that this model cannot deal with the minor spam class effectively. Again, DFFNNs in combination with ensemble learning was more effective than DTs used in RF, Bagging or AdaBoost M1. This suggests that DTs are not that accurate at filtering spam messages, as compared with DNNs.

Table 26: Performance of spam filtering methods in terms of F-score

Method	Enron	SpamAssasin	SMS
FDA+NB	0.975±0.007	0.944±0.009	0.975±0.006
FDA+SVM	0.963±0.014	0.988±0.007	0.985±0.003
IL+C4.5	0.968±0.013	0.964±0.012	0.978±0.003
RF	0.989±0.003	0.974±0.009	0.986±0.004
Voting	0.980±0.004	0.875±0.039	0.989±0.004
CNN	0.997±0.001	0.987±0.009	0.992±0.003
k-NN	0.973±0.007	0.966±0.010	0.961±0.004
AdaBoostM1	0.897±0.010	0.949±0.010	0.938±0.004
Bagging	0.990±0.003	0.968±0.010	0.981±0.004
DFFNN	0.997±0.001	0.990±0.008	0.992±0.003
AdaBoostM1+DFFNN	0.998±0.002	0.990±0.004	0.990±0.002
Bagging+DFFNN	0.999±0.001	0.990±0.006	0.993±0.003
RSS+DFNNN	0.999±0.000	0.991±0.005	0.991±0.004

Method	Hyves	Twitter	Positive hotel reviews	Negative hotel reviews
FDA+NB	0.871±0.038	0.875±0.004	0.858±0.051	0.850±0.035
FDA+SVM	0.856±0.036	0.915±0.027	0.818±0.052	0.840±0.057
IL+C4.5	0.862±0.038	0.945±0.005	0.714±0.041	0.716±0.063
RF	0.880±0.039	0.925±0.006	0.733±0.058	0.709±0.068
Voting	0.875±0.039	0.912±0.012	0.845±0.042	0.861±0.059
CNN	0.886±0.022	0.965±0.000	0.859±0.050	0.856±0.071
k-NN	0.871±0.035	0.933±0.007	0.614±0.079	0.693±0.067
AdaBoostM1	0.887±0.021	0.911±0.002	0.656±0.091	0.735±0.039
Bagging	0.897±0.021	0.940±0.005	0.770±0.067	0.756±0.044
DFFNN	0.911±0.025	0.946±0.004	0.866±0.053	0.888±0.045
AdaBoostM1+DFFNN	0.904±0.027	0.924±0.011	0.854±0.060	0.862±0.033
Bagging+DFFNN	0.916±0.017	0.941±0.006	0.874±0.052	0.905±0.029
RSS+DFNNN	0.914±0.018	0.942±0.005	0.873±0.057	0.895±0.025

9.4 Performance of Spam Filtering Methods in terms of Computational Time

The main limitation of the proposed spam filtering models is that it is substantially more computationally intensive than the other models in terms of training time (Table 27), with average elapsed training time about five times higher than that of Bagging, and about forty times higher than that of DFFNN and CNN. Overall, the proposed models are more computationally complex than other benchmarked methods. Among the proposed methods with ensemble learning, RSS+DFFNN is the least computationally intensive method. On one hand, this finding limits the application of the proposed model in online training mode. On the other hand, the results suggest that the proposed models can be effectively used for static datasets.

Other methods had relatively low training times, especially FDA+SVM, *k*-NN and RF. The ratio of spam and legitimate messages and spam classification domain had little impact on training times, unlike the size of the datasets. The results of the experiments regarding training time are summarized in Table 27.

Table 27: Performance of spam filtering methods in terms of training time

Method	Enron	SpamAssasin	SMS	
FDA+NB	4.162±0.185	2.778±0.062	3.321±0.189	
FDA+SVM	0.237±0.026	0.604±0.124	0.218±0.024	
IL+C4.5	147.13±15.41	32.38±2.21	112.570±7.482	
RF	2.579±0.350	0.528±0.027	7.326±0.250	
Voting	10.49±0.27	1.569±0.188	6.894±0.946	
CNN	41.90±2.71	14.08±0.99	47.850±11.176	
k-NN	0.014±0.020	0.007±0.008	0.007±0.011	
AdaBoostM1	62.30±15.67	36.25±1.65	57.548±3.555	
Bagging	278.07±9.60	103.99±3.34	545.257±159.770	
DFFNN	35.95±5.95	15.79±0.22	44.665±10.239	
AdaBoostM1+DFFNN	1425.1±126.5	1135.6±76.9	1471.906±97.647	
Bagging+DFFNN	1100.0±100.2	488.07±28.37	1663.989±297.369	
RSS+DFNNN	363.51±35.71	198.22±12.32	529.226±35.962	
Method	Hyves	Twitter	Positive hotel reviews	Negative hotel reviews
FDA+NB	0.506±0.056	4.751±0.168	0.459±0.042	0.537±0.095
FDA+SVM	0.076±0.018	2.162±0.089	0.017±0.011	0.023±0.011
IL+C4.5	4.031±0.192	102.10±5.99	7.148±0.609	6.401±0.297
RF	0.478±0.056	168.95±3.16	0.296±0.045	0.270±0.022
Voting	1.947±0.190	16.68±3.07	1.547±0.203	1.564±0.223
CNN	7.548±0.322	6.009±0.189	3.023±1.235	4.559±1.382
k-NN	0.003±0.006	0.087±0.010	0.003±0.006	0.001±0.004
AdaBoostM1	7.037±0.698	55.63±1.88	4.759±0.184	5.260±1.118
Bagging	30.21±1.97	586.41±24.74	34.12±1.03	31.40±2.01
DFFNN	7.412±0.488	21.24±5.41	3.65±55.73	3.729±0.149
AdaBoostM1+DFFNN	400.9±116.0	584.96±19.00	300.97±29.73	279.57±22.05
Bagging+DFFNN	431.54±51.69	235.11±12.20	204.27±21.85	201.42±32.44
RSS+DFNNN	195.16±9.40	109.76±5.71	108.15±10.13	99.31±11.55

To compare the computational time of the proposed models, I also adopted the approach used in previous studies (Chen et al., 2017) and used testing times to demonstrate real-time capacity.

The results in Table 28 show that the proposed models were less time efficient than the other spam filtering models. However, the capacity of the proposed models can be considered to be sufficient for online detection systems because approximately 21,200 messages can be categorized per second, ranging from 9,200 for Hyves to 24,300 for hotel reviews. For example, the average testing times for DFFNN was 41,600 messages/sec, indicating acceptable throughput of the proposed spam detection system irrespective of data size and review domain.

Table 28: Performance of spam filtering methods in terms of testing time

Method	Enron	SpamAssasin	SMS	
FDA+NB	0.912±0.048	0.454±0.023	1.073±0.029	
FDA+SVM	0.000±0.000	0.006±0.008	0.000±0.000	
IL+C4.5	0.006±0.008	0.001±0.004	0.001±0.004	
RF	0.014±0.008	0.001±0.004	0.034±0.014	
Voting	0.656±0.034	0.136±0.026	0.802±0.068	
CNN	2.692±0.372	0.904±0.015	3.139±0.815	
k-NN	6.757±0.913	5.281±0.199	1.907±0.647	
AdaBoostM1	0.000±0.000	0.003±0.006	0.000±0.000	
Bagging	0.004±0.007	0.001±0.004	0.015±0.012	
DFFNN	2.229±0.224	0.978±0.031	2.890±0.439	
AdaBoostM1+DFFNN	33.367±0.506	16.910±0.795	56.314±9.225	
Bagging+DFFNN	33.289±0.246	16.948±0.884	43.539±5.663	
RSS+DFNNN	16.146±0.145	7.807±0.473	19.556±0.438	
Method	Hyves	Twitter	Positive hotel reviews	Negative hotel reviews
FDA+NB	0.178±0.032	9.942±0.100	0.140±0.023	0.145±0.028
FDA+SVM	0.001±0.004	0.012±0.014	0.000±0.000	0.000±0.000
IL+C4.5	0.000±0.000	0.026±0.007	0.000±0.000	0.000±0.000
RF	0.003±0.006	7.856±0.508	0.003±0.006	0.003±0.006
Voting	0.138±0.021	9.922±0.693	0.102±0.015	0.081±0.016
CNN	0.473±0.105	4.479±0.028	0.242±0.008	0.303±0.051
k-NN	0.156±0.052	15.731±0.236	0.253±0.030	0.375±0.040
AdaBoostM1	0.000±0.000	0.015±0.000	0.000±0.000	0.001±0.004
Bagging	0.000±0.000	0.128±0.014	0.001±0.004	0.001±0.004
DFFNN	0.606±0.127	20.229±0.405	0.254±0.014	0.253±0.051
AdaBoostM1+DFFNN	12.604±4.701	333.900±2.047	4.362±0.123	3.812±0.134
Bagging+DFFNN	12.171±3.329	391.379±7.922	4.184±0.158	3.815±0.198
RSS+DFNNN	5.676±0.807	175.956±2.793	2.043±0.088	1.729±0.047

9.5 Performance of Spam Filtering Methods in terms of MC

To evaluate the MC measure, the spam filtering models were tested for different values of misclassification cost ratio in agreement with previous studies (Zhang et al., 2014; Jia and Shang, 2014), λ=1, λ=3, λ=7 and λ=9. Note that for λ=1, MC_1 is the average value of FNR and FPR.

The results of the experiments for MC ratio λ=1 are summarized using average MC as presented in Figures 6-12.

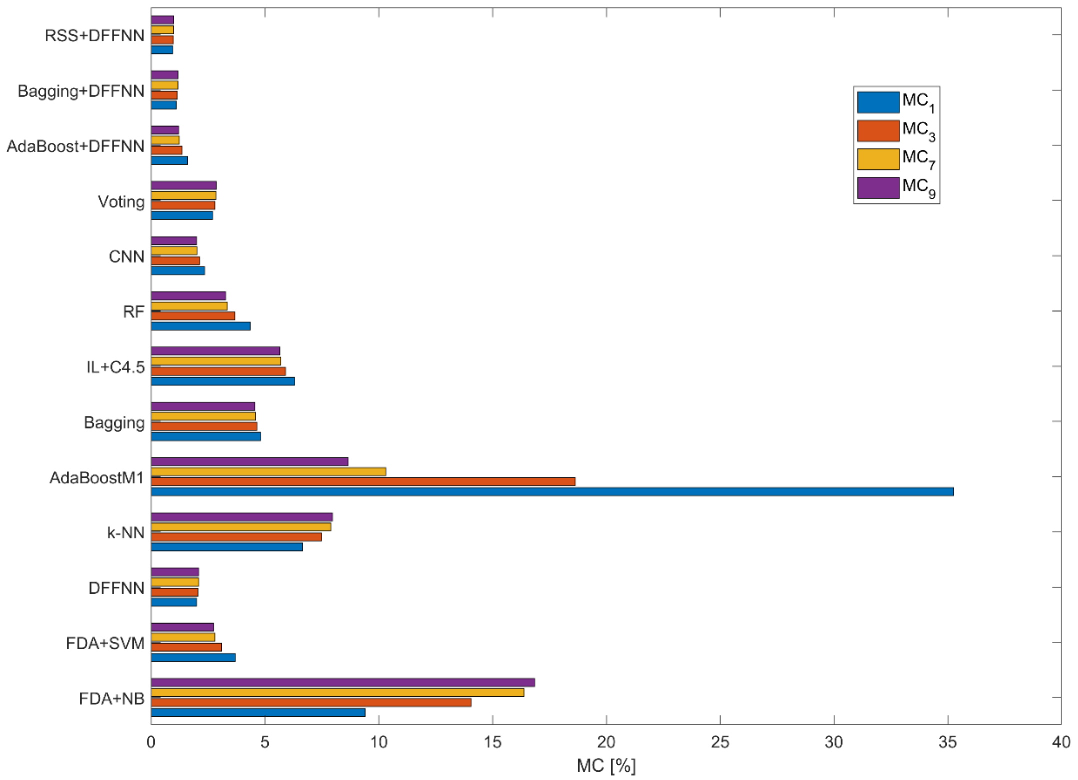

Figure 6: MC for the Enron dataset

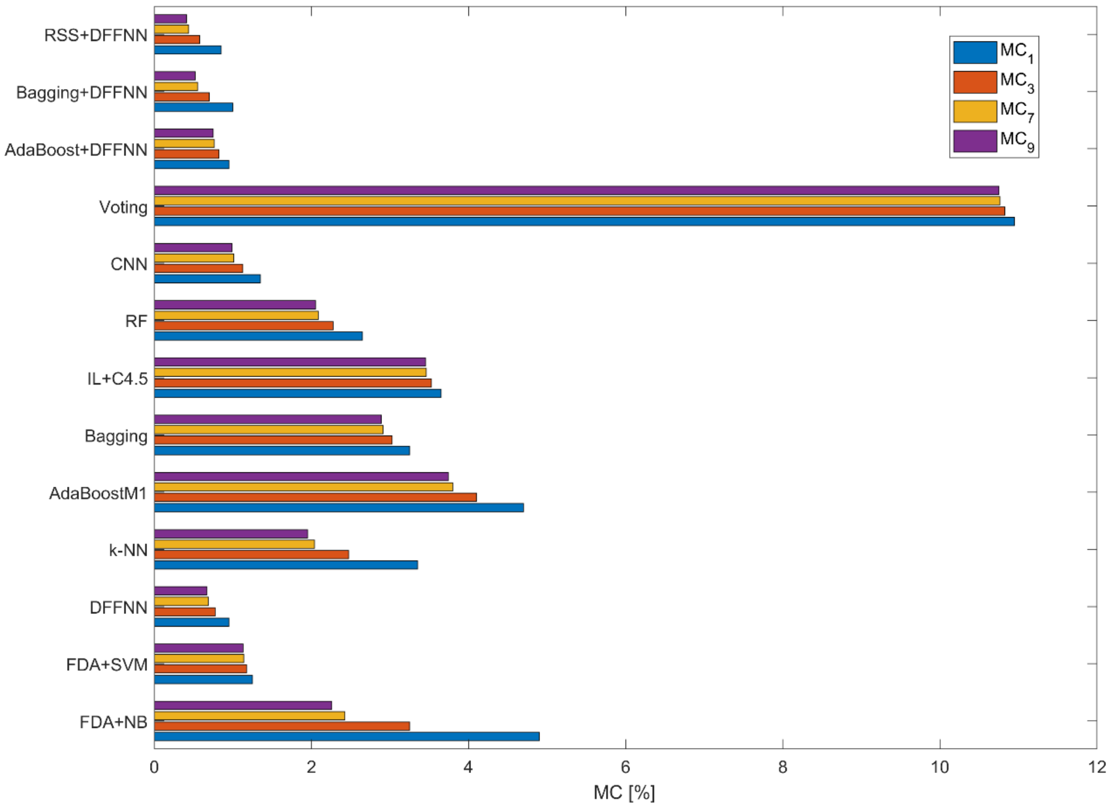

Figure 7: MC for the SpamAssassin dataset

The results for the e-mail datasets in Figure 6 and Figure 7 show that the proposed models performed best in terms of MC, irrespective of λ value. As almost all the methods performed better in terms of FNR, the MC decreased for larger values of λ. Besides the DFFNN with ensemble learning, the DFFNN, CNN and FDA+SVM also performed well.

Figure 8 shows the average MC values for the SMS dataset. Again, the proposed models performed very well and they were outperformed only by CNN for $\lambda = 1$ and $\lambda = 3$. Similarly as for the e-mail datasets, the performance improved for higher values of λ, which represents more realistic scenarios. The worst performance can be seen for the AdaBoost M1 and *k*-NN models. This can be attributed to their poor performance in terms of FNR.

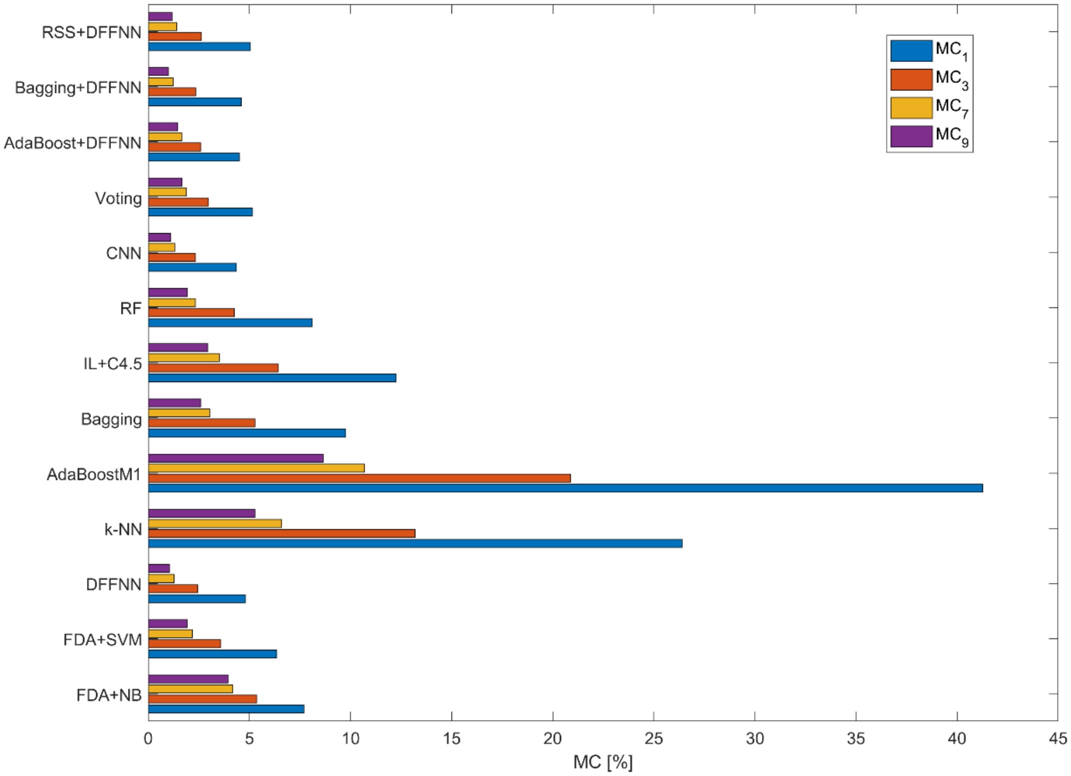

Figure 8: MC for the SMS dataset

Figure 9 and Figure 10 show the results for the social network datasets. Two different results were obtained. For the Hyves dataset, the proposed models performed well and their performance improved with increasing λ value. However, traditional machine learning methods performed better for $\lambda = 7$ and $\lambda = 9$. This can be explained by highly imbalanced social network datasets. Good performance of these methods on the legitimate class in terms of FNR resulted in low MC.

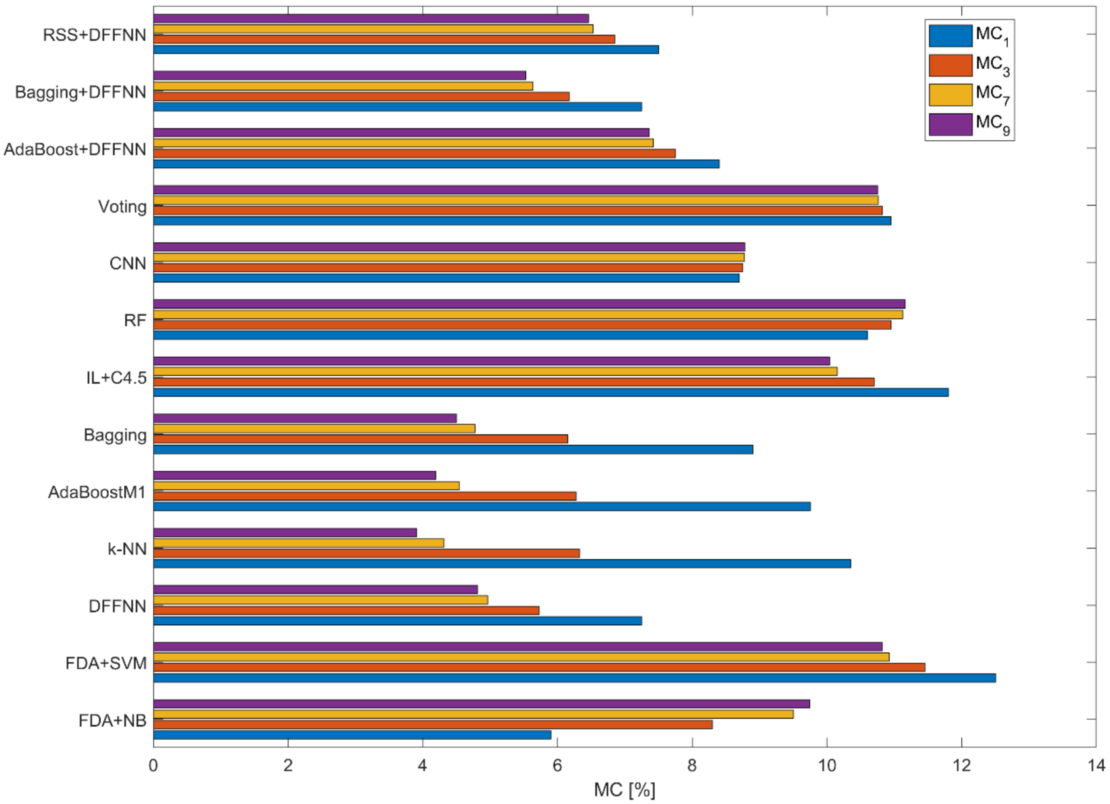

Figure 9: MC for the Hyves dataset

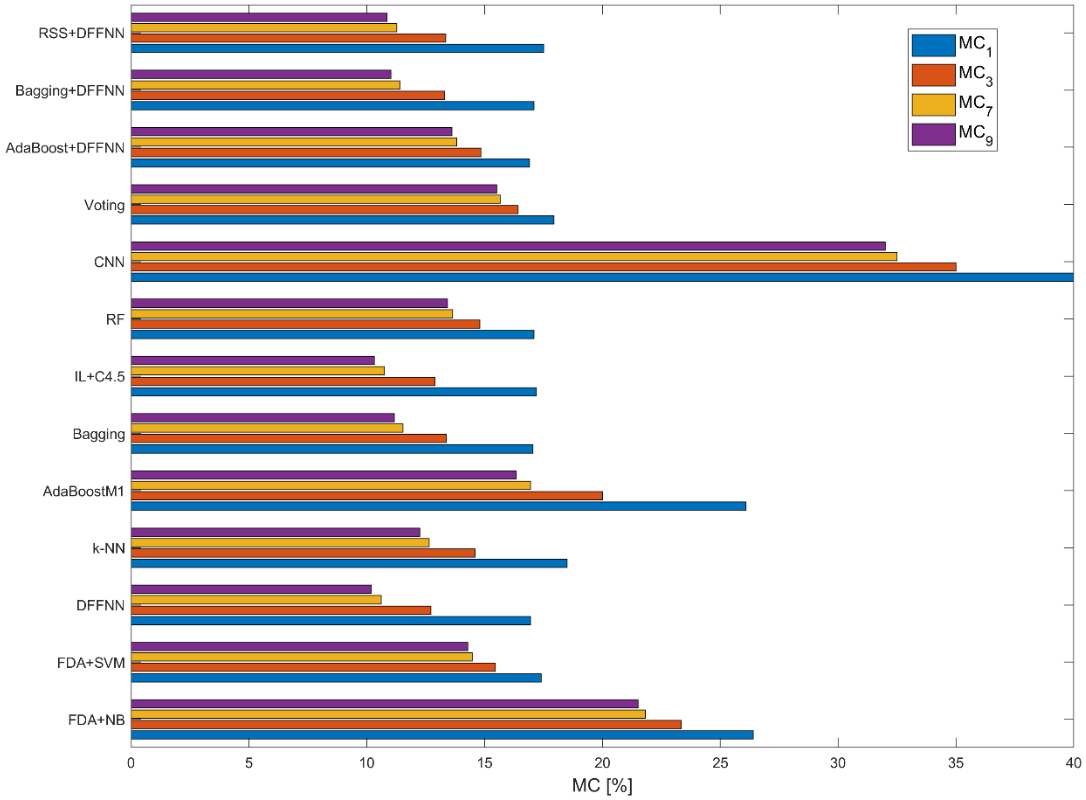

Figure 10: MC for the Twitter dataset

Finally, Figure 11 and Figure 12 present MC for the positive and negative review datasets, respectively. Obviously, the value of λ had no significant effect on MC value, indicating a relatively balanced performance of all the models on both spam and legitimate classes. For all the MC scenarios, the proposed models performed best, with Bagging+DFFNN as the best performer.

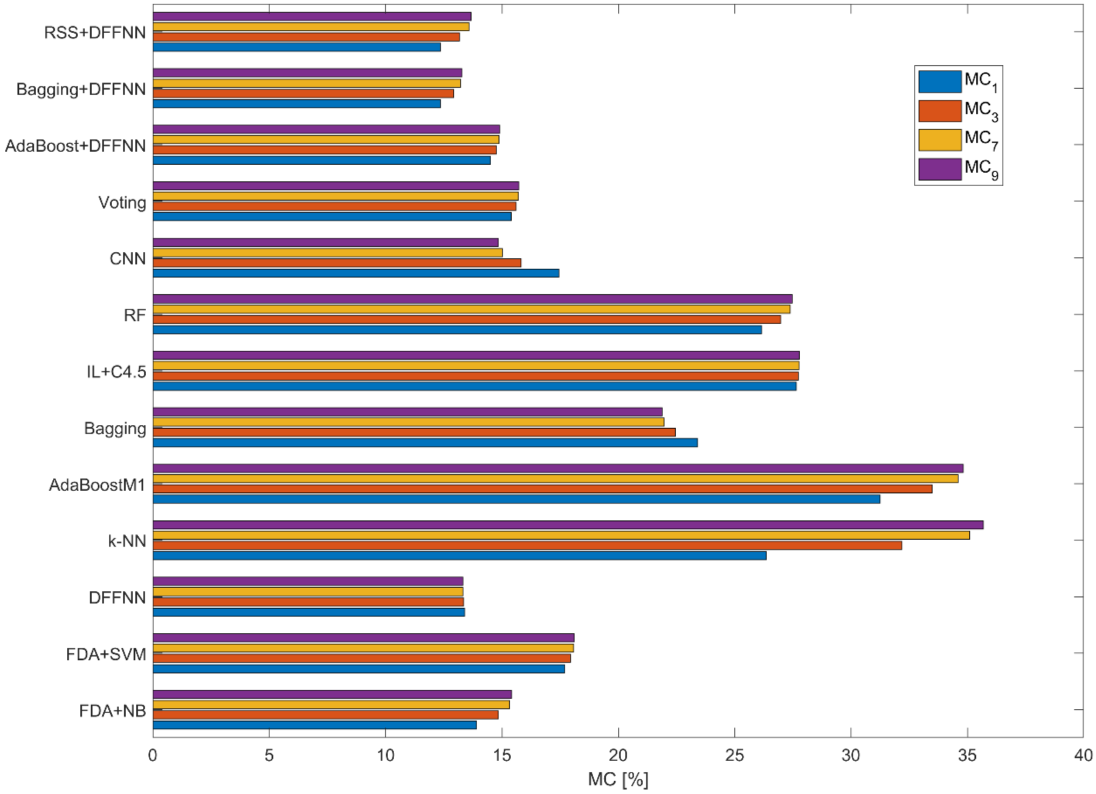

Figure 11: MC for the positive review dataset

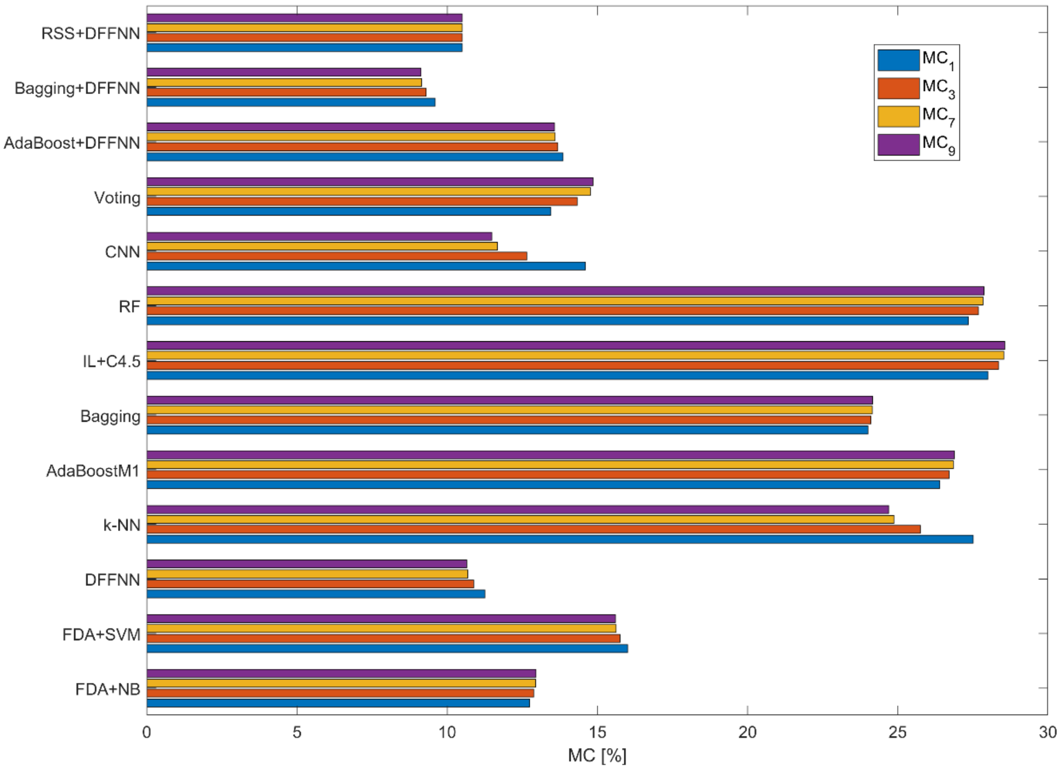

Figure 12: MC for the negative review dataset

9.6 Sensitivity to Feature Selection Methods

To demonstrate the effectiveness of the proposed hybrid model combining the traditional BoW model with word embeddings obtained using Word2Vector (W2V) model, sensitivity to these features was examined in the next set of experiments. Specifically, the experiments were performed separately for the *n*-gram features (BoW) and word embeddings (W2V) using the proposed spam filtering models with ensemble learning, i.e. AdaBoost+DFFNN, Baggin+DFFNN and RSS+DFFNN.

Figures 13-19 clearly show that: (1) word embeddings are less effective than *n*-grams for all the datasets except SpamAssassin in terms of accuracy, (2) models using the *n*-gram performed statistically similar (using two-tailed Student's paired *t*-test at *P*=0.05) as the hybrid models except the SpamAssassin dataset, (3) the effect of machine learning method was not significant, and (4) the hybrid models always performed best, indicating the advantage of combining both sets of features.

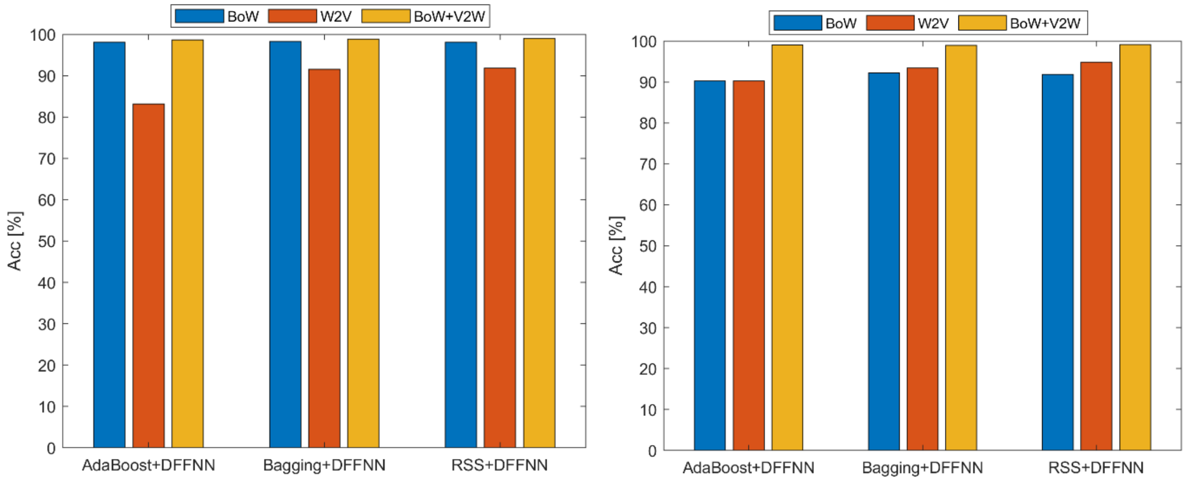

Figure 13: Accuracy of the proposed models for different feature selection methods – e-mail datasets (Enron on the left and SpamAssassin on the right)

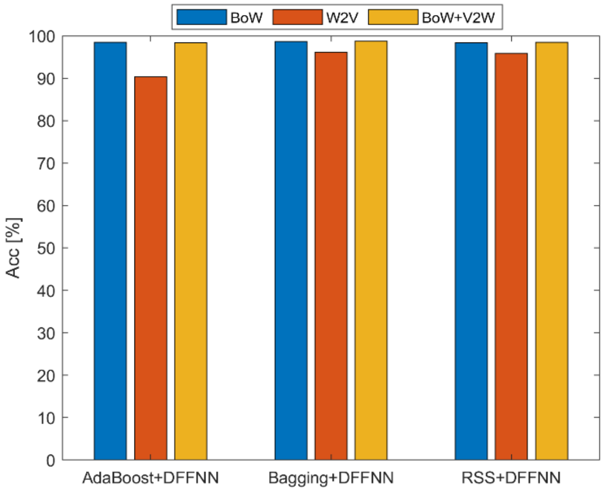

Figure 14: Accuracy of the proposed models for different feature selection methods – SMS dataset

Figure 15 shows the differences for a small (Hyves) and large (Twitter) datasets trained using the Skip-Gram model. Obviously, this model is effective only for large datasets, while its capacity to model word context in a small number of short messages is very weak. By contrast, the sufficient number of *n*-grams was effective for both types of data, irrespective of the type of the ensemble learning method.

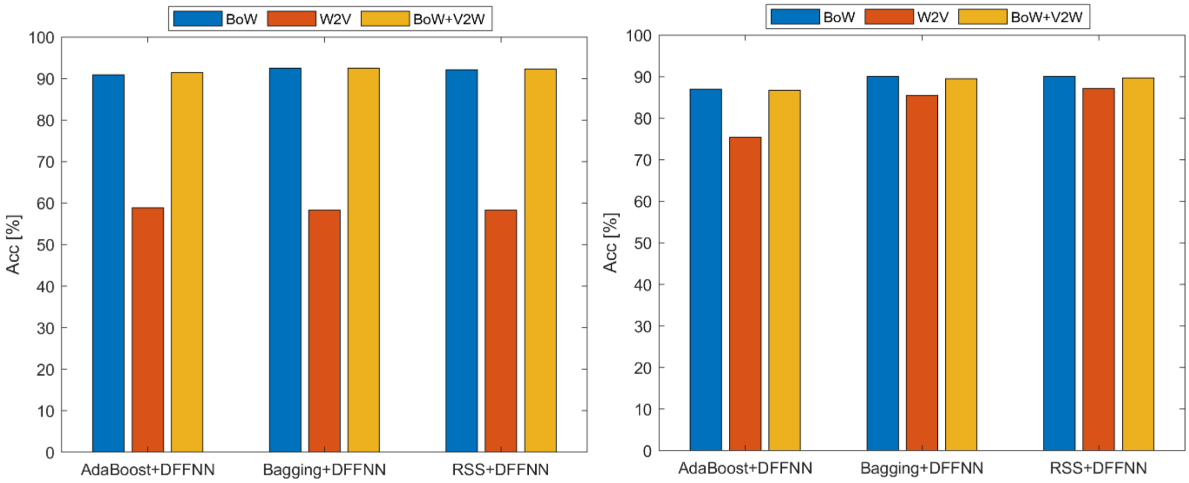

Figure 15: Accuracy of the proposed models for different feature selection methods – social network datasets (Hyves on the left and Twitter on the right)

Figure 16 shows the results for the two hotel review datasets. Thus, we can see that the behavior of the proposed models was similar for both message polarity classes, positive and negative. In general, it was slightly more challenging to accurately classify positive hotel reviews. A possible explanation can be that negative reviews are usually focused on more specific product characteristics and, therefore, easier recognizable from each other.

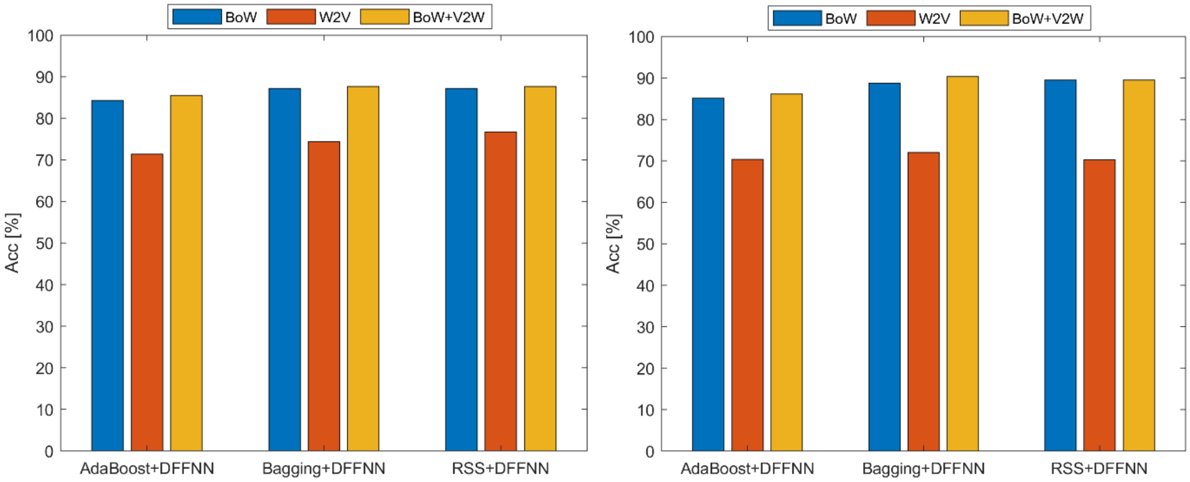

Figure 16: Accuracy of the proposed models for different feature selection methods – hotel review datasets (positive hotel reviews on the left and negative hotel reviews on the right)

9.7 Statistical Comparison of Spam Filtering Methods

In order to compare the performance of the benchmarked spam filtering methods statistically, a nonparametric Friedman test (Garcia et al., 2010) was performed across the seven datasets. This test is based on ranking the methods according to the Friedman statistic. Average ranks were calculated in case of ties. The null hypothesis was tested which states that all the spam filters perform similarly. This test was chosen because the reliability of parametric tests (e.g., data normality) could not be guaranteed for only ten experimental results per dataset.

For the Friedman test, all the previously presented methods were used, including those using BoW and W2V as features (Table 29). The Friedman *P*-value of 2.62E-7 indicates significant differences among the tested spam filtering methods (the chi-square value for 11 degrees of freedom was 52.09). To further compare the results against the best performer, the Holm post-hoc procedure (Garcia et al., 2010) was employed to adjust the significance level.

Among the methods, the Bagging+DFFNN ranked first regarding all the evaluation measures related to prediction accuracy. The baseline methods were significantly outperformed by the proposed models, whereas DFFNN performed statistically similar at *P*=0.05 in terms of accuracy. By contrast, other algorithms performed relatively poor and Adaboost+DFFNN$_{\mathrm{W2V}}$ showed the worst result. The results demonstrate that utilizing ensemble learning help increase accuracy rate. Moreover, the proposed methods benefited from concurrently utilizing *n*-grams and word embeddings.

When it comes to FPR all the proposed models show the best results, while Bagging+DFFNN$_{\mathrm{W2V}}$ along with AdaBoost+DFFNN$_{\mathrm{W2V}}$ performed worst among the tested models. The proposed models also show solid results in terms of FNR. The results demonstrate that Bagging+DFFNN ranked first in terms of both FNR and FPR. Unlike FNR, FDA+NB performed relatively poorly in terms of FPR. Utilizing word embeddings together with *n*-grams helped decrease both these evaluation measures as well. The results also demonstrate that utilizing ensemble learning help improve the ranking of DNNs.

Furthermore, the proposed models demonstrate solid results in terms of F-score and AUC, again with Baggind+DFFNN with the best score. Besides the proposed algorithms, CNN also demonstrates good performance, while the performance of the remaining benchmarked methods was relatively poor. Again, the results confirm that that ensemble algorithms along with the hybrid word representation improved both the F-score and AUC evaluation measures.

Unlike other evaluation criteria, the proposed models were the most computationally intensive with AdaBoost+DFFNN ranking worst. On the one hand, and as expected, *k*-NN ranked first in terms of training time because this algorithm actually requires no training. However, it performs relatively poor when it comes to testing time. On the other hand, FDA+SVM ranked best regarding testing time and relatively well also in terms of training time. The proposed models are more computationally complex due to the DFFNN used as base learner. Moreover, ensemble learning methods increase computational complexity when compared with the single machine learning methods.

Table 29: Nonparametric Friedman test – Average ranking

Method	Acc	FNR	FPR	F-score	AUC	Training time	Testing time
FDA+NB	13.9*	8.4	14.0*	14.2*	12.1*	5.1	12.3
FDA+SVM	11.2	10.2	11.8	11.6	15.6*	2.1	3.1
IL+C4.5	13.1*	16.1*	13.6*	13.6*	17.5*	13.1*	3.3
RF	12.3	12.7	13.6*	12.6	11.8*	5.7	6.9
Voting	11.6	9.3	14.0*	11.9	10.2	6.6	11.1
CNN$_{\text{BoW}}$	14.7*	13.1	14.5*	14.9*	14.2*	4.6	4.8
CNN	9.1	10.2	8.9	5.9	9.6	10.6*	13.3
k-NN	16.0*	15.9*	10.5	15.9*	15.6*	1.0	14.0
AdaBoost M1	16.9*	18.6*	12.1*	16.6*	17.1*	12.6*	3.0
Bagging	11.4	13.9	12.0	11.4	11.5*	16.3*	4.5
DFFNN	4.4	7.2	4.0	4.3	4.8	10.9*	14.1
DFFNN$_{\text{W2V}}$	12.8*	7.9	13.9*	13.0*	11.3*	6.6	3.0
AdaBoost+DFFNN$_{\text{W2V}}$	19.9*	14.9*	19.9*	20.0*	18.3*	7.9	18.9*
AdaBoost+DFFNN$_{\text{BoW}}$	9.4	11.1	10.2	10.0	11.1	18.9*	7.9
AdaBoost+DFFNN	6.6	5.5	8.1	6.7	6.4	20.3*	20.3*
Bagging+DFFNN$_{\text{W2V}}$	16.5*	13.6*	16.1*	15.8*	13.8*	10.3*	18.7*
Bagging+DFFNN$_{\text{BoW}}$	5.1	9.5	5.3	5.4	6.4	18.9*	9.1
Bagging+DFFNN	2.5	4.9	3.8	2.4	2.1	19.4*	20.1*
RSS+DFFNN$_{\text{W2V}}$	15.4*	13.5	14.7*	15.6*	13.5*	8.4	16.1
RSS+DFFNN$_{\text{BoW}}$	6.1	9.4	5.9	6.0	5.9	15.4*	9.6
RSS+DFFNN	2.7	5.3	4.4	3.1	2.2	16.4*	16.9*

Legend: * indicates statistically worse method than the best performer at P=0.05

9.8 Comparison with Previous Studies

To further demonstrate the effectiveness of the proposed spam filtering models, the average accuracy obtained was compared with that of previous studies that examined the same datasets. To ensure fair comparability of the results, Tables 30-32 only report accuracies obtained using 10-fold cross-validation. Similarly, Table 33 presents AUC obtained using 10-fold cross-validation.

Regarding the Enron dataset (Table 30), the best performance thus far reported was achieved by Bagged RF (Shams and Mercer, 2013) and Deep Belief Networks (Tzortzis and Likas, 2007). The results for RF obtained here agree with those from Shams and Mercer (2013). Therefore, I believe that these results suggest that RSS+DFFNN performs better than other methods in terms of accuracy.

Table 30: Comparison of RSS+DFFNN accuracy with the results of previous studies on the Enron dataset

Study	Method	Acc [%]
Tzortzis and Likas (2007)	Deep Belief Networks	97.43
Abi-Haidar and Rocha (2008)	Artificial immune system	90.00
Almeida et al. (2011a)	Multivariate Bernoulli NB	94.79
Uysal and Gunal (2012)	Distinguishing Feature Selector	94.35
Almeida and Yamakami (2012)	Minimum description length	95.56
Shams and Merce (2013)	Bagged RF	97.75
Trivedi and Dey (2013)	Enhanced genetic programming	94.10
Mishra and Thakur (2013)	RF	96.39
Trivedi and Dey (2016b)	Relief + NB	96.30
Hassan (2016)	*k*-means + SVM	97.35
Chhogyal and Nayak (2016)	Natural language toolkit NB	94.70
Sanghani and Kotecha (2016)	Incremental SVM	96.86
Trivedi and Dey (2016a)	Boosted NB + SVM	95.60
Gaurav et al. (2019)	RF	92.30
Gupta et al. (2019)	Ensemble NB and DT	92.40
This study	RSS+DFNNN	**99.05**

For the SpamAssassin dataset, several methods have performed similarly to mine in previous studies, including SVM, AIS, NB, and Boosting, and our comparative results corroborate these findings. However, RSS+DFFNN achieved slightly higher accuracy than prior studies have reported, as presented in Table 31.

Table 31: Comparison of RSS+DFFNN accuracy with the results of previous studies on the SpamAssassin dataset

Study	Method	Acc [%]
Carpinter and Hunt (2006)	Heuristic filter + NB	97.67
Méndez et al. (2007)	SVM	98.53
Fdez-Riverola et al. (2007)	Case-based Reasoning	93.58
Tzortzis and Likas (2007)	Deep Belief Networks	97.50
Yu and Xu (2008)	SVM	97.00
Rozza et al. (2009)	Isotropic PCA	98.89
Zitar and Hamdan (2013)	Genetic optimized AIS	98.92
Trivedi and Dey (2013)	Enhanced genetic programming	98.60
Trivedi and Dey (2016b)	OneR + NB	96.40
Fang (2016)	Maximum entropy + incremental learning	97.87
Shams and Mercer (2016)	Natural language stylometry + AdaBoost	95.70
Trivedi and Dey (2016a)	Boosted NB + SVM	98.60
This study	RSS+DFFNN	**99.14**

Even larger increases in accuracy were achieved in the case of the SMS dataset (Table 32). The SVM proposed in Almeida et al. (2011b) has performed the best so far on this dataset, and the SVM used here reproduced similar results, suggesting that the proposed model is also more effective for SMS spam filtering.

Table 32: Comparison of DBB-RDNN-ReL accuracy with the results of previous studies on the SMS dataset

Study	Method	Acc [%]
Almeida et al. (2011b)	SVM	97.64
Uysal and Gunal (2012)	Distinguishing Feature Selector	97.44
Uysal et al. (2012)	χ^2 filter + probabilistic classifier	90.17
Ahmed et al. (2015)	Apriori + ensemble learning	96.21
Najadat et al. (2016)	Discriminative multinomial NB	96.46
El Boujnouni (2017)	Support Vector Domain Description	89.32
Kaliyar et al. (2018)	SVM	88.00
This study	Bagging+DFFNN	**98.70**

The comparison with previous studies on the Hyves dataset is presented in Table 33. In this case, I report AUC consistent with the comparative study (Bosma et al., 2012). However, note that the results are not fully comparable since I made use of all labelled data whereas the models proposed in Bosma et al. (2012) learned from completely unlabelled or partially labelled data only. Therefore, the results demonstrate that better classification performance can be achieved at the cost of additional manually annotated messages.

Table 33: Comparison of RSS+DFNN with the results of previous studies on the Hyves dataset in terms of AUC

Study	Method	AUC
Bosma et al. (2012)	NB baseline	0.528
Bosma et al. (2012)	Report baseline	0.548
Bosma et al. (2012)	HITS unsupervised	0.767
Bosma et al. (2012)	HITS semi-supervised	0.801
This study	RSS+DFNNN	**0.958**

Table 34 shows the comparison for the two hotel review datasets. Previous studies suggest that SVM is a promising classification method for both hotel review datasets. Obviously, combining these two datasets together only deteriorated the performance by making it difficult to model the review sentiment in addition to their spam characteristics. On the one hand, the proposed model was not capable to beat the existing approaches for the positive review dataset. On the other hand, the best performance so far is reported for the negative review dataset.

Table 34: Comparison of Bagging+DFNN with the results of previous studies on the hotel review datasets

Data	Study	Method	Acc [%]
Positive hotel reviews	Ott et al. (2013)	SVM	89.3
Negative hotel reviews	Ott et al. (2013	SVM	86.0
Pos.+Neg. hotel reviews	Li et al. (2014)	SAGE	81.8
Pos.+Neg. hotel reviews	Shojaee et al. (2013)	SVM, NB	F-score=0.840
Pos.+Neg. hotel reviews	Li et al. (2017b)	SWNN	F-score=0.837
Positive hotel reviews	Fusilier et al. (2015)	NB	F-score=0.882
Negative hotel reviews	Fusilier et al. (2015)	NB	F-score=0.854
Pos.+Neg. hotel reviews	Rout et al. (2017)	LR	83.8
Positive hotel reviews	This study	Bagging+DFFNN	87.63, F-score=0.874
Negative hotel reviews	This study	Bagging+DFFNN	**90.38**, F-score=**0.905**

10. Limitations and Further Research Suggestions

The dissertation thesis was limited to machine learning methods based on supervised learning because all the messages in the datasets were labelled with classes. However, several previous studies also utilized unlabeled reviews and employed methods with unsupervised or semi-supervised learning (Patel and Patel, 2018). Indeed, I expect that including additional unlabeled messages may improve the performance of the proposed models. Collecting additional data is therefore strongly recommended and seems to be a promising approach in future research.

Moreover, all messages in the datasets used for benchmarking were written in English. Besides that, non-alphabetic script languages such as Chinese and Japanese Kanji writing systems may not feasibly benefit from the proposed models due to the nature of the non-alphabetic languages. Tokenization of Chinese and Japanese Kanji scripts can be challenging and would require further research. Therefore, it would be beneficial to investigate whether the proposed models show similar performance for spam datasets from different countries in different languages from different online platforms and whether localization of classification algorithm is required.

Obviously, the proposed models tend to be more computationally intensive (requiring both substantial CPU time and RAM size) than existing traditional algorithms such NB and SVM. Constant computer hardware progress, CPU and RAM are getting more affordable. Moreover, the proposed models can be easily parallelized and executed simultaneously. Therefore, they can benefit from modern advanced CPU and GPU technologies such as multithreading and multi cores processors. High computational expenses make it also more demanding to tackle the problem of concept drift because the trained models must be updated regularly. Further experimentation with concept drift is therefore also strongly recommended for further research.

Another limitation of the proposed model is that author-based features were not fully utilized. Compared with the multi-modal embedding representation proposed in Liu et al. (2019), rich behavior features were neglected, such as the ratio of authors' messages and the rating distribution of an author's reviews. It is therefore recommended that future studies should combine the proposed models with graph-based approaches using authors' metadata. Moreover, the content of the neighboring messages could be utilized in future studies. However, this would require a larger dataset to be collected. Another potential application of this model is to use it to predict spammers in addition to spam messages.

It is also worth to benchmark the proposed models on datasets with multiple classes. In this research, only datasets with binary classes were used. All messages were either labeled as legitimate or spam. In case of positive results, it is possible to further extend use-case scenarios. The results obtained here suggest that the proposed models might have great potential also in other text categorization tasks, such as fake news detection, web-page classification, sentiment classification and so forth. In fact, both *n*-grams and word embeddings can be easily retrained on other large corpuses for alternative domain applications. Therefore, it should be taken into consideration benchmarking the proposed models in other text classification domains.

11. Contributions of the Dissertation Thesis

The aim of the dissertation thesis was to design a machine learning model that would help improve spam filtering performance using ensemble machine learning methods with DNNs to effectively model complex high-dimensional features generated from the message text using *n*-grams and word embeddings. The scientific and application contributions of this dissertation thesis are as follows.

11.1 Scientific Contributions

The scientific contributions of the dissertation thesis include:

- A novel high-dimensional feature selection integrating *n*-gram and Skip-Gram models to model semantic meaning of the messages and the word context. Thus, a high-dimensional document-level representation is obtained. The novelty of this model is the effective exploitation of the word context in messages considering BoW.
- Unlike earlier literature, here I use a Skip-Gram model to obtain the word representation. This model exploits the word context more effectively and thus generates a more generalizable context when compared with the previously used CBOW model.
- A novel spam filtering model based on DFFNN equipped with regularization and ReL units to capture the complex high-dimensional features. Thus, better optimization convergence and resistance to overfitting can be achieved. An important advantage of this model is that no additional dimensionality reduction algorithm is necessary.
- Investigation of the effect of text data preprocessing techniques on the performance of the existing spam filtering methods across different spam filtering domains. Such a comparative study is unique in the existing literature.
- Novel spam filtering models using ensemble algorithms with DFFNN as base classifier. Combining multiple base classifiers helps increase the performance and robustness over the single DFFNN model. In contrast to previous ensemble models using DTs as base classifiers, the proposed approach exploits the advantages of DNNs in handling high-dimensional features and sparse text datasets.

- Benchmark the proposed spam filtering against existing state-of-the-art spam filtering methods. The results demonstrate that the proposed models performed better than the state-of-the-art methods in terms of the most important evaluation criteria.
- For the first time, benchmark datasets were used from multiple spam filtering domains, including e-mail, SMS, social networks and online reviews. This provides strong support to the findings of this dissertation thesis.

11.2 Application Contributions

The application contributions of the dissertation thesis are as follows:

- The proposed spam filtering models can be used in different spam filtering tasks across multiple domains.
- More accurate spam filters may enable to improve the security of business entities in public and private sectors by applying the improved machine learning algorithms in the antispam engine of the e-mail and web security gateways solutions. By implementing ongoing training of the proposed machine learning models, the antispam engine will be adjusted to particular field business entity is working in. Indeed, the results suggest that the spam filtering models can be effectively trained for both the non-personalized e-mail data (SpamAssassin) and the personalized e-mail data (Enron). This is important because spam e-mails decrease work productivity, increase IT support related resources (help desk) and may even result in security incidents.
- More effective spam filtering solutions for cloud services providers and social networks. This will be achieved by utilizing periodically updated massive databases of social media messages available to cloud providers for ongoing training in order to find recent trends (concept drift) in spam generation approaches. This is also important because personal privacy can be threatened and spam messages may by a security threat, in particular when containing links to phishing web sites or servers hosting malware.
- More sophisticated fake review polarity independent filtering on online travel aggregators and shops. This is made possible by taking into consideration semantic meaning of the words by utilizing word embeddings and, therefore, hidden connections between words and deception can be detected. This is also important because fake reviews are becoming a problem due to the fact they may mislead potential buyers which

can result in potential lawsuit against the seller and other adverse effects. Indeed, most marketplaces like Amazon give priority to well-evaluated products (the so-called snowball effect), thus potentially rewarding businesses paying for fake reviews.

- The results of the analysis of preprocessing techniques in different spam filtering domains can be used as recommendations for future spam filtering models in these domains.

Conclusion

In this dissertation thesis, I demonstrated that using high-dimensional document representation obtained using the *n*-gram model and word embeddings together with ensemble learning algorithms with DFFNN as base learners is more accurate than state-of-the-art spam filtering methods. Nine popular spam filtering methods were benchmarked against the three proposed ensemble-based models using seven different datasets from different domains, including e-mail, SMS, social networks and positive and negative online reviews. The results show that the proposed approach based on the ensemble methods demonstrate the best performance in terms of accuracy, FNR, FPR, AUC, F-score and MC, and outperform the state-of-the-art classification methods in most of the evaluation criteria. The results also show that the Bagging algorithm trained with DFFNNs as base classifiers using the combination of word embeddings and *n*-grams as input features achieved the best results for most of the datasets, with a high accuracy on both spam and legitimate classes. This can be attributed to the capacity of Bagging in reducing the risk of overfitting.

The main limitation of the proposed model is that it is more computationally intensive than the compared algorithms. The average testing and especially training CPU time is significantly higher comparing to the state-of-the-art spam filters. On the one hand, this finding limits the application of the proposed model as a spam filter trained online. On the other hand, the results suggest that the proposed model can be effectively used for static datasets. Moreover, constant CPU and GPU computing power performance growth following the Moore's law along with introduction of ASIC chips designed for optimized artificial intelligence computation[11] will help overcome the computation complexity challenge. Implementing the proposed models using a low-level programming language such as assembler will help further lower the computational time in the future.

Moreover, this research also demonstrates the central importance of text preprocessing strategies in detecting spam messages. The results indicate that common patterns can be observed. The number and length of the extracted word segments have major effect on the performance of the classifiers. Therefore, I strongly recommend using the sufficient number of word segments either in the form of bigrams or trigrams. In addition, the non-binary weighting

[11] https://cloud.google.com/tpu

scheme should be applied. The remaining techniques, including removal of stopwords, document normalization and stemming may also improve the classifiers' performance.

To sum up, the combination of complex DFFNNs trained on random subsets of preprocessed high-dimensional data seems to be an effective method for spam filtering in different spam filtering domains.

References

[1] ABI-HAIDAR, A., ROCHA, L. M. Adaptive spam detection inspired by the immune system. In: *Artificial life XI, Proceedings of the 11th International Conference on the Simulation and Synthesis of Living Systems*, 2018, pp. 1–8. doi: 10.1007/978-3- 540-85072-4

[2] ADEWOLE, K. S., ANUAR, N. B., KAMSIN, A., VARATHAN, K. D., RAZAK, S. A. Malicious accounts: dark of the social networks. *Journal of Network and Computer Applications*, 2017b, vol. 79, pp. 41–67. doi: 10.1016/j.jnca.2016.11.030

[3] ADEWOLE, K. S., ANUAR, N. B., KAMSIN, A., SANGAIAH, A. K. SMSAD: A framework for spam message and spam account detection. *Multimedia Tools and Applications*, 2019, vol. 78, pp. 3925–3960. doi: 10.1007/s11042-017-5018-x

[4] AHMED, F., ABULAISH, M. A generic statistical approach for spam detection in Online Social Networks. *Computer Communications*, 2013, vol. 36, no. 10–11, pp. 1120–1129. doi: 10.1016/j.comcom.2013.04.004

[5] AHMED, I., ALI, R., GUAN, D., LEE, Y. K., LEE, S., CHUNG, T. Semi-supervised learning using frequent itemset and ensemble learning for SMS classification. *Expert Systems with Applications*, 2015, vol. 42, no. 3, pp. 1065–1073. doi: 10.1016/j.eswa.2014.08.054

[6] AHMED, H., TRAORE, I., SAAD, S. Detecting opinion spams and fake news using text classification. *Security and Privacy*, 2018, vol. 1, no. 1, e9. doi: 10.1002/spy2.9

[7] AL-JANABI, M., QUINCEY, E., ANDRAS, P. Using supervised machine learning algorithms to detect suspicious URLs in online social networks. In: *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2017, ACM, pp. 1104–1111. doi: 10.1145/3110025.3116201

[8] ALMEIDA, T. A., ALMEIDA, J., YAMAKAMI, A. Spam filtering: how the dimensionality reduction affects the accuracy of Naive Bayes classifiers. *Journal of Internet Services and Applications*, 2011a, vol. 1, vol. 3, pp. 183–200. doi: 10.1007/s13174-010-0014-7

[9] ALMEIDA, T. A., HIDALGO, J. M. G., YAMAKAMI, A. Contributions to the study of SMS spam filtering: new collection and results. In: *Proceedings of the 11th ACM*

Symposium on Document Engineering, 2011b, pp. 259–262. doi: 10.1145/2034691.2034742

[10] ALMEIDA, T. A., SILVA, T. P., SANTOS, I., HIDALGO, J. M. G. Text normalization and semantic indexing to enhance instant messaging and SMS spam filtering. *Knowledge-Based Systems*, 2016, vol. 108, pp. 25–32. doi: 10.1016/j.knosys.2016.05.001

[11] ALMEIDA, T. A., YAMAKAMI, A. Occam's razor-based spam filter. *Journal of Internet Services and Applications*, 2012, vol. 3, no. 3, pp. 245–253. doi: 10.1007/ s13174-012-0067-x

[12] AL-ZOUBI, A. M., FARIS, H., HASSONAH, M. A. Evolving support vector machines using whale optimization algorithm for spam profiles detection on online social networks in different lingual contexts. *Knowledge-Based Systems*, 2018, vol. 153, pp. 91–104. doi: 10.1016/j.knosys.2018.04.025

[13] ANDROUTSOPOULOS, I., KOUTSIAS, J., CHANDRINOS, K. V., SPYROPOULOS, C. D. An experimental comparison of Naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages. In: *Proceedings of the 23rd Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval,* 2000, pp. 160–167. https://doi.org/10.1145/345508.345569

[14] ANTONAKAKI, D., POLAKIS, I., ATHANASOPOULOS, E., IOANNIDIS, S., FRAGOPOULOU, P. Exploiting abused trending topics to identify spam campaigns in Twitter. Social Network Analysis and Mining, 2016, vol. 6, no. 1, 48. doi: 10.1007/s13278-016-0354-9

[15] ARAGAO, M. V., FRIGIERI, E. P., YNOGUTI, C. A., PAIVA, A. P. Factorial design analysis applied to the performance of SMS anti-spam filtering systems. *Expert Systems with Applications*, 2016, vol. 64, pp. 589–604. doi: 10.1016/j.eswa.2016.08.038

[16] ASGHAR, M. Z., ULLAH, A., AHMAD, S., KHAN, A. Opinion spam detection framework using hybrid classification scheme. *Soft Computing*, 2020, vol. 24, pp. 3475–3498. doi: 10.1007/s00500-019-04107-y

[17] ASWANI, R., KAR, A. K., ILAVARASAN, P. V. Detection of spammers in twitter marketing: a hybrid approach using social media analytics and bio inspired computing. *Information Systems Frontiers*, 2018, vol. 20, pp. 515–530. doi: 10.1007/s10796-017-9805-8

[18] BARBADO, R., ARAQUE, O., IGLESIAS, C. A. A framework for fake review detection in online consumer electronics retailers. *Information Processing & Management*, 2019, vol. 56, no. 4, pp. 1234–1244. doi: 10.1016/j.indmarman.2019.08.003

[19] BARUSHKA, A., HAJEK, P. Spam filtering using regularized neural networks with rectified linear units. In: Adorni, G., Cagnoni, S., Gori, M., Maratea, M. (eds.) *Conference of the Italian Association for Artificial Intelligence. Lecture Notes in Computer Science*, Springer, Cham, 2016, vol. 10037, pp. 65–75. doi: 10.1007/978-3-319-49130-1_6

[20] BARUSHKA, A., HAJEK, P. Spam filtering in social networks using regularized deep neural networks with ensemble learning. In: Iliadis, L., Maglogiannis, I., Plagianakos, V. (eds.) *IFIP International Conference on Artificial Intelligence Applications and Innovations*, AIAI 2018, Springer, Cham, 2018a, vol. 519, pp. 38–49. doi: 10.1007/978-3-319-92007-8_4

[21] BARUSHKA, A., HAJEK, P. Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks. *Applied Intelligence*, 2018b, vol. 48, no. 10, pp. 3538–3556. doi: 10.1007/s10489-018-1161-y

[22] BARUSHKA, A., HAJEK, P. Spam detection on social networks using cost-sensitive feature selection and ensemble-based regularized deep neural networks. *Neural Computing and Applications*, 2019a, pp. 1-19. doi: 10.1007/s00521-019-04331-5

[23] BARUSHKA, A., HAJEK, P. Review spam detection using word embeddings and deep neural networks. In: MacIntyre, J., Maglogiannis, I., Iliadis, L., Pimenidis, E. (eds.) Artificial Intelligence Applications and Innovations. AIAI 2019. *IFIP Advances in Information and Communication Technology*, vol. 559, Springer, Cham, 2019b, pp. 340–350. doi: 10.1007/978-3-030-19823-7_28

[24] BENEVENUTO, F., MAGNO, G., RODRIGUES, T., ALMEIDA, V. Detecting spammers on twitter. In: *7th Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference* (CEAS 2010), 2010, pp. 1–12.

[25] BHAT, S. Y., ABULAISH, M. Community-based features for identifying spammers in online social networks. In: *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, IEEE, 2013, pp 100–107. doi: 10.1145/2492517.2492567

[26] BHOWMICK, A., HAZARIKA, S. M. E-mail spam filtering: A review of techniques and trends. In: Kalam A, Das S, Sharma K (eds.) *Advances in Electronics, Communication and Computing*. Lecture Notes in Electrical Engineering, Springer, Singapore, vol. 443, 2018, pp. 583–590. doi: 10.1007/978-981-10-4765-7_61

[27] BINDU, P. V., MISHRA, R., THILAGAM, P. S. Discovering spammer communities in twitter. *Journal of Intelligent Information Systems*, 2018, vol. 51, pp. 503–527. doi: 10.1007/s10844-017-0494-z

[28] BOSMA, M., MEIJ, E., WEERKAMP, W. A framework for unsupervised spam detection in social networking sites. In: Baeza-Yates R et al. (eds.) *European Conference on Information Retrieval*. Springer, Berlin, Heidelberg, 2012, pp. 364–375. doi: 1007/978-3-642-28997-2_31

[29] BREIMAN, L. Bagging predictors. *Machine Learning*, 1996, vol. 24, no. 2, pp. 123–140. doi: 10.1007/BF00058655

[30] BREIMAN, L. Random forests. *Machine Learning*, 2001, vol. 45, no. 1, pp. 5–32. doi: 10.1023/A:1010933404324

[31] BRIGHTLOCAL. *Local consumer review survey 2018*, 2018. Accessed 30 November 2019, available at: https://www.brightlocal.com/research/local-consumer-review-survey/.

[32] CAO, C., CAVERLEE, J. Detecting spam urls in social media via behavioral analysis. In: *European Conference on Information Retrieval*, Springer, Cham, 2015, pp. 703–714. doi: 10.1007/978-3-319-16354-3_77

[33] CARPINTER, J., HUNT, R. Tightening the net: a review of current and next generation spam filtering tools. *Computers & Security*, 2006, vol. 25, no. 8, pp. 566–578. doi: 10.1016/j.cose.2006.06.001

[34] CARUANA, G., LI, M. A survey of emerging approaches to spam filtering. *ACM Computing Surveys*, 2008, vol. 44, no. 2, pp. 1–27. doi: 10.1145/2089125.2089129

[35] CHAN, P. P., YANG, C., YEUNG, D. S., NG, W. W. Spam filtering for short messages in adversarial environment. *Neurocomputing*, 2015, vol. 155, pp. 167–176. doi: 10.1016/j.neucom.2014.12.034

[36] CHANDY, R., GU, H. Identifying spam in the iOS app store. In: *Proceedings of the 2nd Joint WICOW/AIRWeb Workshop on Web Quality*. ACM, 2012, pp. 56–59. doi: 10.1145/ 2184305.2184317.

[37] CHEN, C., WANG, Y., ZHANG, J., XIANG, Y., ZHOU, W., MIN, G. Statistical features-based real-time detection of drifted twitter spam. *IEEE Transactions on Information Forensics and Security*, 2017a, vol. 12, no. 4, pp. 914–925. doi: 10.1109/TIFS.2016.2621888

[38] CHEN, W., YEO, C. K., LAU, C. T., LEE, B. S. A study on real-time low-quality content detection on Twitter from the users' perspective. *PloS One*, 2017b, vol. 12, no. 8, pp. e0182487. doi: 10.1371/journal.pone.0182487

[39] CHHOGYAL, K., NAYAK, A. An empirical study of a simple Naive Bayes classifier based on ranking functions. In: *Australasian Joint Conference on Artificial Intelligence*. Springer, 2016, pp. 324–331. doi: 10.1007/978-3-319-50127-7 27

[40] CHOUDHARY, N., JAIN, A. K. Towards filtering of SMS spam messages using machine Learning Based Technique. In: Singh, D., Raman, B., Luhach, A., Lingras, P. (eds.) *Advanced Informatics for Computing Research*. Communications in Computer and Information Science, vol. 712, Springer, Singapore, 2017, pp. 18–30. doi: 10.1007/978-981-10-5780-9_2

[41] CHU, Z., WIDJAJA, I., WANG, H. Detecting social spam campaigns on twitter. In: *International Conference on Applied Cryptography and Network Security*, Springer, Berlin, Heidelberg, 2012, pp. 455–472. doi: 10.1007/978-3-642-31284-7_27

[42] CORMACK, G. V. Email spam filtering: a systematic review. *Foundations and Trends® in Information Retrieval*, 2006, vol. 1, no. 4, pp. 335–455. https://doi.org/10.1561/150 0000006

[43] CORMACK, G. V., HIDALGO, J. M. G., SÁNZ, E. P. Feature engineering for mobile (SMS) spam filtering. In: *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, 2007, pp. 871–872. doi: 10.1145/1277741.1277951

[44] CRAWFORD, M., KHOSHGOFTAAR, T.M., PRUSA, J.D., RICHTER, A.N., AND AL NAJADA, H. Survey of review spam detection using machine learning techniques. *Journal of Big Data*, 2015, vol. 2, no. 1, pp. 1-23. doi: 10.1186/s40537-015-0029-9

[45] DADA, E. G., BASSI, J. S., CHIROMA, H., ADETUNMBI, A. O., AJIBUWA, O. E. Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon*, 2019, vol. 5, no. 6, pp. e01802. doi: 10.1016/j.heliyon.2019.e01802

[46] DELANY, S. J., BUCKLEY, M., GREENE, D. SMS spam filtering: methods and data. *Expert Systems with Applications*, 2012, vol. 39, no. 10, pp. 9899–9908. doi: 10.1016/j.eswa.2012.02.053

[47] DIALE, M., CELIK, T., VAN DER WALT, C. Unsupervised feature learning for spam email filtering. *Computers & Electrical Engineering*, 2019, vol. 74, pp. 89–104. doi: 10.1016/j.compeleceng.2019.01.004

[48] DRUCKER, H., WU, D., VAPNIK, V. Support vector machines for spam categorization. *IEEE Transactions on Neural Networks*, 1999, vol. 10, no. 5, pp. 1048–1054. doi: 10.1109/72.788645

[49] DUTTA, S., GHATAK, S., DEY, R., DAS, A. K., GHOSH, S. Attribute selection for improving spam classification in online social networks: a rough set theory-based approach. *Social Network Analysis and Mining*, 2018, vol. 8, no. 7, pp. 1–16. doi: 10.1007/s13278-017-0484-8

[50] EL BOUJNOUNI, M. SMS spam filtering using N-gram method, information gain metric and an improved version of SVDD classifier. *Journal of Engineering Science & Technology Review*, 2017, vol. 10, no. 1, pp. 131–137

[51] ELMURNGI, E., GHERBI, A. An empirical study on detecting fake reviews using machine learning techniques. In: *7th Int. Conf. on Innovative Computing Technology (INTECH)*. IEEE, 2017, pp. 107–114. doi: 10.1109/INTECH.2017. 8102442.

[52] FANG, A. Applications of the maximum entropy principle in spam email classification. *Journal of Residuals in Science & Technology*, 2016, vol. 13, no. 6, pp. 1–4. doi: 10.12783/issn.1544-8053/13/6/1

[53] FAWCETT, T. In vivo spam filtering: a challenge problem for KDD. *ACM SIGKDD Explorations Newsletter*, 2003, vol. 5, no. 2, pp. 140–148. doi: 10.1145/980972.980990

[54] FDEZ-RIVEROLA, F., IGLESIAS, E. L., DIAZ, F., MENDEZ, J. R., CORCHADO, J. M. Spamhunting: an instance-based reasoning system for spam labelling and filtering. *Decision Support Systems*, 2007 43(3):722–736. doi: 10.1016/j.dss.2006.11.012

[55] FLOYD, K., FRELING, R., ALHOQAIL, S., CHO, H. Y., FRELING, T. How online product reviews affect retail sales: A meta-analysis. *Journal of Retailing*, 2014, vol. 90, no. 2, pp. 217–232. doi: 10.1016/j.jretai.2014.04.004.

[56] FREUND, Y., SCHAPIRE, R. Experiments with a new boosting algorithm. In: *Thirteenth International Conference on Machine Learning*, San Francisco, 1996, pp. 148–156.

[57] FREUND, Y., SCHAPIRE, R., ABE, N. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 1999, vol. 14, no. 5, pp. 771–780.

[58] FUSILIER, D. H., MONTES-Y-GÓMEZ, M., ROSSO, P., CABRERA, R. G. Detection of opinion spam with character n-grams. In: *International Conference on Intelligent Text Processing and Computational Linguistics*, Springer, Cham, 2015, pp. 285–294. doi: 10.1007/978-3-319-18117-2_21

[59] GARCIA, S., FERNANDEZ, A., LUENGO, J., HERRERA, F. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power. *Information Sciences*, 2010, vol. 180, no. 10, pp. 2044–2064. doi: 10.1016/j.ins.2009.12.010

[60] GAURAV, D., TIWARI, S. M., GOYAL, A., GANDHI, N., ABRAHAM, A. Machine intelligence-based algorithms for spam filtering on document labeling. *Soft Computing*, 2019, pp. 1–14. doi: 10.1007/s00500-019-04473-7

[61] GEORGE, P., VINOD, P. Composite email features for spam identification. In: *Cyber Security, Advances in Intelligent Systems and Computing*, vol. 729, Springer, 2018, pp. 281–289. doi: 10.1007/978-981-10-8536-9_28

[62] GHAI, R., KUMAR, S., PANDEY, A. C. Spam detection using rating and review processing method. In: *Smart Innovations in Communication and Computational Sciences*. Springer, Singapore, 2019, pp. 189–198. doi: 10.1007/978-981-10-8971-8_18

[63] GOGOGLOU, A., THEODOSIOU, Z., KOUNOUDES, T., VAKALI, A., MANOLOPOULOS, Y. Early malicious activity discovery in microblogs by social bridges detection. In: *2016 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, IEEE, Limassol, 2016, pp. 132–137. doi: 10.1109/ISSPIT.2016.7886022

[64] GUPTA, M., BAKLIWAL, A., AGARWAL, S., MEHNDIRATTA, P. A comparative study of spam SMS detection using machine learning classifiers. In: *2018 Eleventh International Conference on Contemporary Computing (IC3)*, IEEE, 2018, pp 1–7. 10.1109/IC3.2018.8530469

[65] GUPTA, V., MEHTA, A., GOEL, A., DIXIT, U., PANDEY, A. C. Spam detection using ensemble learning. In: *Harmony Search and Nature Inspired Optimization Algorithms, Advances in Intelligent Systems and Computing*, Vol. 74, Springer, 2019, pp. 661–668. 10.1007/978-981-13-0761-4_63

[66] GUZELLA, T., CAMINHAS, W. A review of machine learning approaches to spam filtering. *Expert Systems with Applications*, 2009, vol. 36, no. 7, pp. 10206–10222. doi: 10.1016/j.eswa. 2009.02.037

[67] HAGENAU, M., LIEBMANN, M., NEUMANN, D. Automated news reading: stock price prediction based on financial news using context-capturing features. *Decision Support Systems*, 2013, vol. 55, no. 3, pp. 685–697. https://doi.org/10.1016/j.dss.2013.02.006

[68] HARRIS, C. Detecting deceptive opinion spam using human computation. In: *Workshops at the 26th AAAI Conference on Artificial Intelligence*. AAAI, 2012, pp. 87–93.

[69] HASSAN, D. Investigating the effect of combining text clus- tering with classification on improving spam email detection. In: Madureira, A., Abraham, A., Gamboa, D., Novais, P. (eds.) *International Conference on Intelligent Systems Design and Applications*. Springer, Cham, 2016, pp. 99–107. doi: 10.1007/978-3-319- 53480-0 10

[70] HEALY, M., DELANY, S., ZAMOLOTSKIKH, A. An assessment of case base reasoning for short text message classification, In: Creaney, N. (ed.) *Proceedings of the 15th Irish Conference on Artificial Intelligence & Cognitive Science* (AICS'05), 2005, pp. 257–266.

[71] HENNING, J. L. SPEC CPU2006 benchmark descriptions. *ACM SIGARCH Computer Architecture News*, 2006, vol. 34, no. 4, pp. 1–17. doi: 10.1145/1186736.1186737

[72] HEYDARI, A., ALI TAVAKOLI, M., SALIM, N., HEYDARI, Z. Detection of review spam: A survey. *Expert Systems with Applications*, 2015, vol. 42, no. 7, pp. 3634–3642. doi: 10.1016/j.eswa.2014. 12.029

[73] HIDALGO, J., BRINGAS, G., SANZ, E., GARCIA, F. Content based SMS spam filtering. In: *Proceeding of the ACM Symposium on Document Engineering*, 2006, pp. 107–114. doi: 10.1145/1166160.1166191

[74] HINTON, G., SRIVASTAVA, N., KRIZHEVSKY, A., SUTSKEVER, I., SALAKHUTDINOV, R. *Improving neural networks by preventing co-adaptation of feature detectors*, 2012, arXiv:1207.0580.

[75] HO, T. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998, vol. 20, no. 8, pp. 832–844. doi: 10.1109/34.709601

[76] HOANCA, B. How good are our weapons in the spam wars? *IEEE Technology and Society Magazine*, 2006, vol. 25, no. 1, pp. 22–30. doi: 10.1109/MTAS. 2006.1607720

[77] JAITLY, N., HINTON, G. Learning a better representation of speech soundwaves using restricted Boltzmann machines. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 5884–5887. doi: 10.1109/ICASSP. 2011.5947700

[78] JIA, X., SHANG, L. Three-way decisions versus two-way decisions on filtering spam email. In: *Transactions on Rough Sets XVIII*, 2014, Springer, Berlin, Heidelberg, pp. 69–91. doi: 10.1007/978-3-662-44680-5_5

[79] JIN, X., LIN, C., LUO, J., HAN, J. A data mining-based spam detection system for social media networks. In: *Proceedings of the VLDB Endowment*, 2011, vol. 4, no. 12, pp. 1458–81461.

[80] JINDAL, N., LIU, B. Analyzing and detecting review spam. In: *7th IEEE Int. Conf. on Data Mining*. ICDM 2007, IEEE, 2007, pp. 547–552. doi: 10.1109/ICDM.2007.68.

[81] KALIYAR, R. K., NARANG, P., GOSWAMI, A. SMS spam filtering on multiple background datasets using machine learning techniques: A novel approach. In: *2018 IEEE 8th International Advance Computing Conference (IACC)*, IEEE, 2018, pp. 59–65. doi: 10.1109/IADCC.2018.8692097

[82] KAUR, R., SINGH, S., KUMAR, H. Rise of spam and compromised accounts in online social networks: A state-of-the-art review of different combating approaches. *Journal of Network and Computer Applications*, 2018, vol. 112, pp. 53–88. doi: 10.1016/j.jnca.2018.03.015

[83] KAYA, Y., ERTUGRUL, O. F. A novel approach for spam email detection based on shifted binary patterns. *Security and Communication Networks,* 2016, vol. 9, no. 10, pp. 1216–1225. doi: 10.1002/sec.1412

[84] KENNEDY, S., WALSH, N., SLOKA, K., MCCARREN, A., FOSTER, J. Fact or factitious? Contextualized opinion spam detection. In: *Proceedings of the 57th Annual*

Meeting of the Association for Computational Linguistics: Student Research Workshop. ACL, 2019, pp. 344–350. doi: 10.18653/v1/P19-2048.

[85] KHORSHIDPOUR, Z., HASHEMI, S., HAMZEH, A. Evaluation of random forest classifier in security domain. *Applied Intelligence*, 2017, vol. 47, no. 2, 558–569. doi: 10.1007/s10489-017-0907-2

[86] KIM, Y. (2014) *Convolutional neural networks for sentence classification*. arXiv:1408.5882

[87] KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model. *Selection International Joint Conference on Artificial Intelligence*, 1995. doi: 10.1.1.48.529

[88] KOPRINSKA, I., POON, J., CLARK, J., CHAN, J. Learning to classify e-mail. *Information Sciences*, 2007, vol. 177, no. 10, pp. 2167–2187. doi: 10.1016/j.ins. 2006.12.005

[89] LAI, C. An empirical study of three machine learning methods for spam filtering. *Knowledge-Based Systems*, 2007, vol. 20, no. 3, pp. 249–254. doi: 10.1016/j.knosys. 2006.05.016

[90] LAORDEN, C., UGARTE-PEDRERO, X., SANTOS, I., SANZ, B., NIEVES, J., BRINGAS, P. G. Study on the effectiveness of anomaly detection for spam filtering. *Information Sciences*, 2014, vol. 277, pp. 421–444. doi: 10.1016/j.ins. 2014.02.114

[91] LAU, R. Y., LIAO, S. Y., KWOK, R. C. W., XU, K., XIA, Y., LI, Y. Text mining and probabilistic language modeling for online review spam detecting. *ACM Transactions on Management Information Systems*, 2011, vol. 2, no. 4, pp. 1–30. doi: 10.1145/2070710. 2070716.

[92] LE, Q., MIKOLOV, T. Distributed representations of sentences and documents. In: *International Conference on Machine Learning*, 2014, vol. 32, pp. 1188–1196.

[93] LECUN, Y., BOTTOU, L., BENGIO, Y., HAFFNER, P. Gradient- based learning applied to document recognition. *Proceedings of the IEEE*, 1998, vol. 86, no. 11, pp. 2278–2324. https://doi.org/10.1109/5.726791

[94] LEE, H. Y., KANG, S. S. Word embedding method of SMS messages for spam message filtering. In: *2019 IEEE International Conference on Big Data and Smart Computing (BigComp),* IEEE, 2019, pp. 1–4. doi: 10.1109/BIGCOMP.2019.8679476

[95] LEE, K., CAVERLEE, J., WEBB, S. Uncovering social spammers: social honeypots+ machine learning. In: *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, 2010, pp. 435–442. doi: 10.1145/1835449.1835522

[96] LEE, K., EOFF, B. D., CAVERLEE, J. Seven months with the devils: A long-term study of content polluters on Twitter. In: *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, 2011, pp. 185–192.

[97] LEE, S., KIM, J. Warningbird: A near real-time detection system for suspicious urls in twitter stream. *IEEE Transactions on Dependable and Secure Computing*, 2013, vol. 10, no. 3, pp. 183–195. doi: 10.1109/TDSC.2013.3

[98] LI, H., FEI, G., WANG, S., LIU, B., SHAO, W., MUKHERJEE, A., SHAO, J. Bimodal distribution and co-bursting in review spam detection. In: 26th Int. Conf. on World Wide Web. ACM, 2017b, pp. 1063–1072. doi: 10.1145/3038912.3052582

[99] LI, H., CHEN Z, MUKHERJEE A, LIU B, SHAO J (2015) Analyzing and detecting opinion spam on a large-scale dataset via temporal and spatial patterns. In: *9th Int. AAAI Conf. on Web and Social Media (ICWSM 2015)*. AAAI, pp. 634–637.

[100] LI, J., OTT, M., CARDIE, C., HOVY, E. Towards a general rule for identifying deceptive opinion spam. In: *Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics*. ACL 1, 2014, pp. 1566–1576. doi: 10.3115/v1/P14-1147

[101] LI, F., HUANG, M., YANG, Y., AND ZHU, X. Learning to identify review spam. In: *Int. Joint Conf. on Artificial Intelligence (IJCAI 2011)*, 2011, pp. 2488-2493.

[102] LI, L., QIN, B., REN, W., AND LIU, T. Document representation and feature combination for deceptive spam review detection. *Neurocomputing*, 2017a, vol. 254, pp. 33–41. doi: 10.1016/j.neucom.2016.10.080.

[103] LI, Q., JIN, Z., WANG, C., ZENG, D. D. Mining opinion summarizations using convolutional neural networks in Chinese microblogging systems. *Knowledge-Based Systems*, 2016, vol. 107, pp. 289–300. doi: 10.1016/j.knosys.2016.06.017

[104] LILLEBERG, J., ZHU, Y., AND ZHANG, Y. Support vector machines and word2vec for text classification with semantic features. In: *2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, IEEE, 2015, pp. 136–140. doi: 10.1109/ICCI-CC.2015.7259377

[105] LIM, E. P., NGUYEN, V. A., JINDAL, N., LIU, B., LAUW, H. W. Detecting product review spammers using rating behaviors. In: *19th ACM Int. Conf. on Information and Knowledge Management*. ACM, 2010, pp. 939–948. doi: 10.1145/1871437.1871557

[106] LIU, C., WANG, G. Analysis and detection of spam accounts in social networks. In: *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*. IEEE, 2016, pp. 2526–2530. doi: 10.1109/CompComm.2016.7925154

[107] LIU, W., WANG, T. Online active multi-field learning for efficient email spam filtering. *Knowledge and Information Systems*, 2012, vol. 33, no. 1, pp. 117–136. 10.1007/s10115-011-0461-x

[108] LIU, Y., PANG, B., WANG, X. Opinion spam detection by incorporating multimodal embedded representation into a probabilistic review graph. *Neurocomputing*, 2019, vol. 366, pp. 276–283. doi: 10.1016/j.neucom.2019. 08.013

[109] LIU, Y., WANG, Y., FENG, L., ZHU, X. Term frequency combined hybrid feature selection method for spam filtering. *Pattern Analysis and Applications*, 2016, vol. 19, no. 2, pp. 369–383. doi: 10.1016/j.asoc.2016.06.043

[110] LIU, Y., PANG, B. A unified framework for detecting author spamicity by modeling review deviation. *Expert Systems with Applications*, 2018, vol. 112, pp. 148–155. doi: 10.1016/j.eswa.2018.06.028.

[111] MAAS, A. L., HANNUN, A. Y., NG, A. Y. Rectifier nonlinearities improve neural network acoustic models. In: *Proceedings of the 30th International Conference on Machine Learning*, vol. 30, 2013, pp. 1–6.

[112] MARTINEZ-ROMO, J., ARAUJO, L. Detecting malicious tweets in trending topics using a statistical analysis of language. *Expert Systems with Applications*, 2013, vol. 40, no. 8, pp. 2992–3000. doi: 10.1016/j.eswa.2012.12.015

[113] MÉNDEZ, J., CORZO, B., GLEZ-PEÑA, D., FDEZ-RIVEROLA, F., DÍAZ, F. Analyzing the performance of spam filtering methods when dimensionality of input vector changes. In: *Machine Learning and Data Mining in Pattern Recognition*, Springer, Berlin, Heidelberg, 2007, pp. 364–378. doi: 10.1007/978-3-540-73499-4_28

[114] METSIS, V., ANDROUTSOPOULOS, I., PALIOURAS, G. Spam filtering with Naive Bayes - which Naive Bayes? In: *Third Conference on Email and Antispam (CEAS)*, 2006, pp. 27–28. doi: 10.1.1.61.5542

[115] MIKOLOV, T., SUTSKEVER, I., CHEN, K., CORRADO, G. S., DEAN, J. Distributed representations of words and phrases and their compositionality. In: *Advances in Neural Information Processing Systems*, NIPS, vol. 26, 2013, pp. 3111–3119.

[116] MILLER, Z., DICKINSON, B., DEITRICK, W., HU, W., WANG, A. H. Twitter spammer detection using data stream clustering. *Information Sciences*, 2014, vol. 260, pp. 64–73. doi: 10.1016/j.ins.2013.11.016

[117] MOKRI, M. A. E. S., HAMOU, R. M., AMINE, A. A new bio inspired technique based on octopods for spam filtering. *Applied Intelligence*, 2019, vol. 49, pp. 3425–3435. doi: 10.1007/s10489-019-01463-y

[118] MUKHERJEE, A., VENKATARAMAN, V., LIU, B., GLANCE, N. What yelp fake review filter might be doing?. In: *7th Int. AAAI Conf. on Weblogs and Social Media*. AAAI, 2013, pp. 409–418.

[119] MURATA, N., YOSHIZAWA, S., AMARI, S. I. Network information criterion-determining the number of hidden units for an artificial neural network model. *IEEE Transactions on Neural Networks*, 1994, vol. 5, no. 6, pp. 865-872.

[120] NAJADAT, H., ABDULLA, N., ABOORAIG, R., NAWASRAH, S. Spam detection for mobile short messaging service using data mining classifiers. *International Journal of Computer Science and Information Security*, 2016, vol. 14, no. 8, pp. 511–517.

[121] NEXGATE. *2013 State of Social Media Spam Research Report*, 2013. Accessed 10 January 2020, available at: https://go.proofpoint.com/rs/309-RHV-619/images/Nexgate-2013-State-of-Social-Media-Spam-Research-Report.pdf.

[122] OBIED, A., ALHAJJ, R. Fraudulent and malicious sites on the web. *Applied Intelligence*, 2009, vol. 30, no. 2, pp. 112–120. https://doi.org/10.1007/s10489- 007-0102-y

[123] OTT, M., CARDIE, C., HANCOCK, J. Estimating the prevalence of deception in online review communities. In: *21st Int. Conf. on World Wide Web*. ACM, 2012, pp. 201–210. doi: 10.1145/2187836.2187864

[124] OTT, M., CARDIE, C., HANCOCK, J. Negative deceptive opinion spam. In: *2013 Conf. of the North American Chapter of the Assoc. for Computational Linguistics: Human Language Technologies*. ACL, 2013, pp. 497–501.

[125] PANDEY, A. C., RAJPOOT, D. S. Spam review detection using spiral cuckoo search clustering method. *Evolutionary Intelligence*, 2019, vol. 12, no. 2, pp. 147–164. doi: 10.1007/s12065-019-00204-x

[126] PATEL, N. A., PATEL, R. A survey on fake review detection using machine learning techniques. In: *2018 4th Int. Conf. on Computing Communication and Automation (ICCCA)*. IEEE, 2018, pp. 1–6. doi: IEEE.10.1109/ CCAA.2018.8777594

[127] PÉREZ-DÍAZ, N., RUANO-ORDÁS, D., FDEZ-RIVEROLA, F., MÉNDEZ, J. R. SDAI: An integral evaluation methodology for content-based spam filtering models. *Expert Systems with Applications*, 2012, vol. 39, no. 16, pp. 12487–12500. doi: 10.1016/j.eswa. 2012.04.064

[128] QUINLAN, J. R. Improved use of continuous attributes in c4.5. *Journal of Artificial Intelligence Research*, 1996, vol. 4, pp. 77–90. doi: 10.1613/jair.279

[129] RAYANA, S., AKOGLU, L. Collective opinion spam detection: Bridging review networks and metadata. In: *21th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 985–994. doi: 10.1145/2783258.2783370

[130] REN, Y., JI, D. Neural networks for deceptive opinion spam detection: An empirical study. *Information Sciences*, 2017, vol. 385, pp. 213–224. doi: 10.1016/j.ins.2017.01.015

[131] ROUT, J. K., DALMIA, A., CHOO, K. K. R., BAKSHI, S., JENA, S. K. Revisiting semi-supervised learning for online deceptive review detection. *IEEE Access*, 2017, vol. 5, pp. 1319–1327. doi: 10.1109/ACCESS.2017.2655032

[132] ROZZA, A., LOMBARDI, G., CASIRAGHI, E. Novel IPCA-based classifiers and their application to spam filtering. In: *9th International Conference on Intelligent Systems Design and Applications, ISDA'09*. IEEE, 2009, pp. 797–802. doi: 10.1109/ISDA.2009.21

[133] SAHAMI, M., DUMAIS, S., HECKERMAN, D., HORVITZ, E. A Bayesian approach to filtering junk e-mail. In: *Learn for Text Categorization*, papers from the 1998 workshop, vol. 62, 1998, pp. 98– 105. doi: 10.1.1.48.1254

[134] SANGHANI, G., KOTECHA, K. Personalized spam filtering using incremental training of support vector machine. In: *International conference on computing, analytics and security trends (CAST)*. IEEE, 2016, pp. 323–328. doi: 10.1109/CAST.2016.7914988

[135] SCHMIDHUBER, J. Deep learning in neural networks: An overview. *Neural Networks*, 2015, vol. 61, pp. 85–117. doi: 10.1016/j.neunet.2014.09.003

[136] SCULLEY, D., WACHMAN, G. M. Relaxed online SVMs for spam filtering. In: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2007, pp. 415–422.

[137] SEBASTIANI, F. Machine learning in automated text categorization. *ACM Computing Surveys (CSUR)*, 2002, vol. 34, no. 1, pp. 1–47. doi: 10.1145/505282.505283

[138] SEDHAI, S., SUN, A. Semi-supervised spam detection in Twitter stream. *IEEE Transactions on Computational Social Systems*, 2018, vol. 5, no. 1, pp. 169–175. doi: 10.1109/TCSS.2017.2773581

[139] SHAMS, R., MERCER, R. E. Personalized spam filtering with natural language attributes. In: *12th International Conference on Machine Learning and Applications (ICMLA),* vol 2. IEEE, 2013, pp. 127–132. doi: 10.1109/ICMLA.2013.117

[140] SHAMS, R., MERCER, R. E. Supervised classification of spam emails with natural language stylometry. *Neural Computing and Applications*, 2016, vol. 27, no. 8, pp. 2315–2331. doi: 10.1007/s00521-015-2069-7

[141] SHEN, H., LI, Z. Leveraging social networks for effective spam filtering. *IEEE Transactions on Computers*, 2014, vol. 63, no. 11, pp. 2743–2759. doi: 10.1109/TC.2013.152

[142] SHEN, H., MA, F., ZHANG, X., ZONG, L., LIU, X., LIANG, W. Discovering social spammers from multiple views. *Neurocomputing*, 2017, vol. 225, pp. 49–57. doi: 10.1016/j.neucom.2016.11.013

[143] SHEU, J. J., CHU, K. T., LI, N. F., LEE, C. C. An efficient incremental learning mechanism for tracking concept drift in spam filtering. *PloS One*, 2017, vol. 12, no. 2, pp. e0171518. doi: 10.1371/journal.pone.0171518

[144] SHOJAEE, S., MURAD, M. A. A., AZMAN, A. B., SHAREF, N. M., NADALI, S. Detecting deceptive reviews using lexical and syntactic features. In: *13th Int. Conf. on Intelligent Systems Design and Applications*. IEEE, 2013, pp. 53–58. doi: 10.1109/ISDA.2013.6920707

[145] SOHRABI, M. K., KARIMI, F. A feature selection approach to detect spam in the Facebook social network. *Arabian Journal for Science and Engineering*, 2018, vol. 43, no. 2, pp. 949–958. doi: 10.1007/s13369-017-2855-x

[146] SOLIMAN, A., GIRDZIJAUSKAS, S. Adaptive graph-based algorithms for spam detection in social networks. *KTH Royal Institute of Technology*, 2016, diva2:998690.

[147] SONG, J., LEE, S., KIM, J. Spam filtering in twitter using sender-receiver relationship. In: *International Workshop on Recent Advances in Intrusion Detection*, Springer, Berlin, Heidelberg, 2011, pp. 301–317.

[148] SONG, L., LAU, R. Y. K., KWOK, R. C. W., MIRKOVSKI, K., DOU, W. Who are the spoilers in social media marketing? Incremental learning of latent semantics for social spam detection. *Electronic Commerce Research*, 2017, vol. 17, no. 1, pp. 51–81. doi: 10.1007/s10660-016-9244-5

[149] STATISTA. *Number of Facebook users worldwide 2008-2018*. 2018a, accessed at 30 November 2019, available at: https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/

[150] STATISTA. *Twitter: number of monthly active users 2010-2018*. 2018b, accessed at 30 November 2019, available at: https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/

[151] STATISTA. *Global spam volume as percentage of total e-mail traffic from January 2014 to September 2019*. 2019a, accessed at 10 January 2020, available at: https://www.statista.com/statistics/420391/spam-email-traffic-share/

[152] STATISTA. *Number of Global smartphone penetration rate as share of population from 2016 to 2020*. 2019b, accessed at 10 January 2020, available at: https://www.statista.com/statistics/203734/global-smartphone-penetration-per-capita-since-2005/

[153] STRINGHINI, G., KRUEGEL, C., VIGNA, G. Detecting spammers on social networks. In: *Proceedings of the 26th Annual Computer Security Applications Conference*. ACM, 2010, pp. 1–9.

[154] SUN, C., DU, Q., TIAN, G. Exploiting product related review features for fake review detection. *Mathematical Problems in Engineering*, 2016, vol. 2016, pp. 1–7. doi: 10.1155/2016/4935792

[155] TALBOT, D. Where spam is born. *MIT Technology Review*, 2008, available at: https://www.technologyreview.com/s/409944/where-spam-is-born/

[156] TANG, X., QIAN, T., YOU, Z. Generating behavior features for cold-start spam review detection. In: *Int. Conf. on Database Systems for Advanced Applications*. Springer, Cham, 2019, pp. 324–328. doi: 10.1007/978-3-030-18590-9_38

[157] THE TIMES. *A third of TripAdvisor reviews are fake' as cheats buy five stars*. 2018, accessed 22 January 2019, available at: https://www.thetimes.co.uk/article/hotel-and-caf-cheats-are-caught-trying-to-buy-tripadvisor-stars-027fbcwc8.

[158] THOMAS, K., GRIER, C., SONG, D., PAXSON, V. Suspended accounts in retrospect: an analysis of twitter spam. In: *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*. ACM, 2011, pp. 243–258.

[159] TRIVEDI, S. K., DEY, S. An enhanced genetic programming approach for detecting unsolicited emails. In: *IEEE 16th International Conference on Computational Science and Engineering (CSE)*, 2013, pp. 1153–1160. doi: 10.1109/CSE.2013.171

[160] TRIVEDI, S. K., DEY, S. A comparative study of various supervised feature selection methods for spam classification. In: *Proceedings of the 2nd International Conference on Information and Communication Technology for Competitive Strategies*. ACM, 2016a, pp. 64. doi: 10.1145/2905055.2905122

[161] TRIVEDI, S. K., DEY, S. A combining classifiers approach for detecting email spams. In: *30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*. IEEE, 2016b, pp. 355–360. doi: 10.1109/WAINA.2016.127

[162] TZORTZIS, G., LIKAS, A. Deep belief networks for spam filtering. In: *19th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2007*, vol 2. IEEE, 2007, pp. 306–309. doi: 10.1109/ICTAI.2007.65

[163] UYSAL, A. K., GUNAL, S., A novel probabilistic feature selection method for text classification. *Knowledge-Based Systems*, 2012, vol. 36, pp. 226–235. doi: 10.1016/j.knosys.2012.06.005

[164] UYSAL, A. K., GUNAL, S., ERGIN, S., GUNAL, E. S. A novel framework for SMS spam filtering. In: *2012 International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*. IEEE, 2012, pp. 1–4. https://doi.org/10.1109/INISTA.2012.6246947

[165] UYSAL, A. K., GUNAL, S. The impact of preprocessing on text classification. *Information Processing & Management*, 2014, vol. 50, no. 1, pp. 104–112. doi: 10.1016/j.ipm.2013.08. 006.

[166] VIDANAGAMA, D. U., SILVA, T. P., KARUNANANDA, A. S. Deceptive consumer review detection: a survey. *Artificial Intelligence Review*, 2020, vol. 53, 1323–1352. doi: 10.1007/s10462-019-09697-5.

[167] VYAS, T., PRAJAPATI, P., GADHWAL, S. A survey and evaluation of supervised machine learning techniques for spam e-mail filtering. In: *IEEE international conference on electrical, computer and communication technologies (ICECCT)*. IEEE, 2015, pp. 1–7. doi: 10.1109/ICECCT.2015.7226077

[168] WANG, A. H. Don't follow me: Spam detection in Twitter. In: *Proceedings of the 2010 International Conference on Security and Cryptography (SECRYPT)*. IEEE, 2010, pp. 1–10.

[169] WANG, G., XIE, S., LIU, B., PHILIP, S. Y. Review graph based online store review spammer detection. In: *11th Int. Conf. on Data mining (ICDM 2011)*. IEEE, 2011, pp. 1242–1247. doi: 10.1109/ICDM.2011.124.

[170] WATCHARENWONG, N., SAIKAEW, K. Spam detection for closed Facebook groups. In: *2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE)*. IEEE, 2017, pp. 1–6. doi: 10.1109/JCSSE.2017.8025914

[171] WATKINS, A., TIMMIS, J. Artificial immune recognition system (AIRS): an immune-inspired supervised learning algorithm. *Genetic Programming and Evolvable Machines*, 2004, vol. 5, no. 3, pp. 291–317. doi: 10.1023/B:GENP.0000030197.83685.94

[172] WEI, C. P., CHEN, H. C., CHENG, T. H. Effective spam filtering: a single-class learning and ensemble approach. *Decision Support Systems*, 2008, vol. 45, no. 3, pp. 491–503. doi: 10.1016/j.dss.2007.06.010

[173] WU, F., SHU, J., HUANG, Y., YUAN, Z. Co-detecting social spammers and spam messages in microblogging via exploiting social contexts. *Neurocomputing*, 2016, vol. 201, pp. 51–65. doi: 10.1016/j.neucom.2016.03.036

[174] XIE, S., WANG, G., LIN, S., YU, P. S. Review spam detection via temporal pattern discovery. In: *18th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*. ACM, 2012, pp. 823–831. doi: 10.1145/2339530.2339662.

[175] XUE, H., WANG, Q., LUO, B., SEO, H., LI, F. Content-aware trust propagation toward online review spam detection. *Journal of Data and Information Quality (JDIQ)*, 2019, vol. 11, no. 3, pp. 11. doi: 10.1145/3305258.

[176] YANG, C., HARKREADER, R., GU, G. Empirical evaluation and new design for fighting evolving twitter spammers. *IEEE Transactions on Information Forensics and Security*, 2013, vol. 8, no. 8, pp. 1280–1293. doi: 10.1109/TIFS.2013.2267732

[177] YE, J., KUMAR, S., AKOGLU, L. Temporal opinion spam detection by multivariate indicative signals. In: *10th Int. AAAI Conf. on Web and Social Media (ICWSM 2016)*. AAAI, 2016, pp. 743–746.

[178] YILMAZ, C. M., DURAHIM, A. O. SPR2EP: A semi-supervised spam review detection framework. In: *2018 IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 2018, pp. 306–313. doi: 10.1109/ASONAM.2018.8508314.

[179] YU, B., XU, Z. B. A comparative study for content- based dynamic spam classification using four machine learning algorithms. *Knowledge-Based Systems*, 2008, vol. 21, no. 4, pp. 355–362. https://doi.org/10.1016/j.knosys.2008.01.001

[180] YU, D., CHEN, N., JIANG, F., FU, B., QIN, A. Constrained NMF-based semi-supervised learning for social media spammer detection. *Knowledge-Based Systems*, 2017, vol. 125, pp. 64–73. doi: 10.1016/j.knosys.2017.03.025

[181] ZENG, Z. Y., LIN, J. J., CHEN, M. S., CHEN, M. H., LAN, Y. Q., LIU, J. L. A review structure-based ensemble model for deceptive review spam. *Information*, 2019, vol. 10, no. 7, pp. 243. doi: 10.3390/info10070243

[182] ZHANG, L., ZHU, J., YAO, T. An evaluation of statistical spam filtering techniques. *ACM Transactions on Asian Language Information Processing*, 2004, vol. 3, no. 4, pp. 243–269. doi: 10.1.1.109.7685

[183] ZHANG, Y., WANG, S., PHILLIPS, P., JI, G. Binary PSO with mutation operator for feature selection using decision tree applied to spam detection. *Knowledge-Based Systems*, 2014, vol. 64, no. 22–31. doi: 10.1016/j.knosys.2014.03.015

[184] ZHENG, X., ZENG, Z., CHEN, Z., YU, Y., RONG, C. Detecting spammers on social networks. *Neurocomputing*, 2015, vol. 159, pp. 27–34. doi: 10.1016/j.neucom.2015.02.047

[185] ZHENG, X., ZHANG, X., YU, Y., KECHADI, T., RONG, C. ELM-based spammer detection in social networks. *The Journal of Supercomputing*, 2016, vol. 72, no. 8, pp. 2991–3005. doi: 10.1007/s11227-015-1437-5

[186] ZHOU, B., YAO, Y., LUO, J. A three-way decision approach to email spam filtering. In: *Canadian Conference on Artificial Intelligence*, *Lecture Notes in Computer Science*, vol. 6085. Springer, 2010, pp. 28–39. doi: 10.1007/978-3-642-13059-5_6

[187] ZHOU, B., YAO, Y., LUO, J. Cost-sensitive three-way email spam filtering. *Journal of Intelligent Information Systems*, 2014, vol. 42, no. 1, pp. 19–45. doi: 10.1007/s10844-013-0254-7

[188] ZITAR, R. A., HAMDAN, A. Genetic optimized artificial immune system in spam detection: a review and a model. *Artificial Intelligence Review*, 2013, vol. 40, no. 3, pp. 305–377. doi: 10.1007/s10462-011-9285-z

Publications of the Student

Journal papers

[1] BARUSHKA, A., HÁJEK, P. Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks. *Applied Intelligence*, 2018, vol. 48, no. 10, pp. 3538–3556. doi: 10.1007/s10489-018-1161-y. IF: 2.882

[2] BARUSHKA, A., HÁJEK, P. Spam detection on social networks using cost-sensitive feature selection and ensemble-based regularized deep neural networks. *Neural Computing and Applications*, 2019, pp. 1-19. doi: 10.1007/s00521-019-04331-5. IF: 4.664

[3] HÁJEK, P., BARUSHKA, A., MUNK, M. Fake consumer review detection using deep neural networks integrating word embeddings and emotion mining. *Neural Computing and Applications*, 2020, pp. 1-16, doi: 10.1007/s00521-020-04757-2. IF: 4.664

Conference papers

[4] BARUSHKA, A., HÁJEK, P. Spam filtering using regularized neural networks with rectified linear units. In: Adorni, G., Cagnoni, S., Gori, M., Maratea, M. (eds.) *Conference of the Italian Association for Artificial Intelligence. Lecture Notes in Computer Science*, Springer, Cham, 2016, vol. 10037, pp. 65–75. doi: 10.1007/978-3-319-49130-1_6

[5] BARUSHKA, A., HÁJEK, P. Spam filtering in social networks using regularized deep neural networks with ensemble learning. In: Iliadis, L., Maglogiannis, I., Plagianakos, V. (eds.) *IFIP International Conference on Artificial Intelligence Applications and Innovations*, AIAI 2018, Springer, Cham, 2018, vol. 519, pp. 38–49. doi: 10.1007/978-3-319-92007-8_4

[6] BARUSHKA, A., HÁJEK, P. Review spam detection using word embeddings and deep neural networks. In: MacIntyre, J., Maglogiannis, I., Iliadis, L., Pimenidis, E. (eds.) Artificial Intelligence Applications and Innovations. AIAI 2019. *IFIP Advances in Information and Communication Technology*, vol. 559, Springer, Cham, 2019, pp. 340–350. doi: 10.1007/978-3-030-19823-7_28

[7] HÁJEK, P., BARUSHKA, A. Integrating sentiment analysis and topic detection in financial news for stock movement prediction. In: *Proceedings of the 2nd International*

Conference on Business and Information Management, 2018, 158–162, doi: 10.1145/3278252.3278267

[8] BARUSHKA, A., HÁJEK, P. The effect of text preprocessing strategies on detecting fake consumer reviews. In: *International Conference on E-Business and Internet 2019*, 2019, in press.

[9] HÁJEK, P., BARUSHKA, A. A comparative study of machine learning methods for detection of fake online consumer reviews. In: *International Conference on E-Business and Internet 2019*, 2019, in press.