Supervised and Unsupervised Applications of Natural Language Processing on Free Text Towards Tackling Scams

Zeya Lwin Tun
201378539

Supervised by Dr Daniel Birks and Dr Leonid Bogachev

Submitted in accordance with the requirements for the
module MATH5872M: Dissertation in Data Science and Analytics
as part of the degree of

Master of Science in Data Science and Analytics

The University of Leeds, School of Mathematics

September 2020

The candidate confirms that the work submitted is his own and that appropriate credit has been given where reference has been made to the work of others.

Abstract

Scams are becoming increasingly prevalent and a cause of concern globally. In Singapore, scams made up 27.0% of overall crimes in 2019, compared to 17.5% in 2018. In the first half of 2020, a total of S$82 million was cheated from victims, almost twice the amount in the same period of 2019. Besides immediate financial losses, victims of scams also suffer from longer-term emotional and psychological effects. Despite efforts by authorities, victims continue to fall prey, owing partly to more sophisticated means used by scammers. There is therefore a strong need to increase the understanding of scams and how they can be prevented.

The research in this dissertation aims to achieve this by drawing insights from others' scam experiences shared on 'Scam Alert', a Singapore-based website aimed at promoting scam awareness. More specifically, this research harnesses the hidden potential of free text in these scam reports using machine learning and Natural Language Processing (NLP) methods towards the following research goals: finding scam reports with similar modus operandi, extracting common characteristics from similar scam reports and classifying scam reports.

In pursuit of these research goals, this dissertation presents novel applications of machine learning and NLP on free text in scam reports in two areas: supervised and unsupervised. In supervised application, deep learning techniques are used for multi-class classification of scam reports. Given class imbalance in the data, text augmentation techniques and the Synthetic Minority Over-sampling Technique are explored. In addition, the efficacy of using pre-trained Global Vectors (GloVe) word embeddings is examined. Results show that the Long Short-Term Memory model trained without GloVe word embeddings on a dataset balanced with text augmentation outperformed the rest.

In unsupervised application, the concept of vector semantics is leveraged using doc2vec models to encode scam reports as document embeddings. To evaluate doc2vec models, a new framework known as normalised Similarity-Dissimilarity Quotient (SDQ) is introduced. Normalised SDQ assesses a doc2vec model's ability to infer document embeddings that can recognise similar and dissimilar reports from sets of pre-identified scam reports. Using normalised SDQ, the most optimal doc2vec model is found to be the model trained with 150 epochs, 50-dimensional embeddings and the Distributed Memory Model of Paragraph Vector algorithm.

Findings from both supervised and unsupervised applications lay the foundation for the development of tools towards achieving the research goals. It is envisioned that these tools will sharpen the sense-making capabilities of law enforcement authorities in better understanding how scams operate and in identifying intervention points where scams can be disrupted. With such insights, public education and engagement efforts can be more tailored and effective. Additionally, these tools can nurture a stronger sense of awareness and guardianship within the society. After all, a discerning public is the strongest defence against scams. All analyses and modelling underlying our research are reproducible using Python code available at the following Github repository: https://github.com/zeyalt/msc-dissertation-final.

Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
ASC	Anti-Scam Centre
BERT	Bidirectional Encoder Representations from Transformers
Bi-LSTM	Bidirectional Long Short-Term Memory
BOW	Bag-of-Words
CBOW	Continuous Bag-of-Words
COVID-19	Coronavirus Disease 2019
CPU	Central Processing Unit
DOM	Document Object Model
EDA	Exploratory Data Analysis
GloVe	Global Vectors
GPU	Graphics Processing Unit
IQR	Inter-Quartile Range
KNN	*k*-Nearest Neighbours
LDA	Latent Dirichlet Allocation
LR	Logistic Regression
LSTM	Long Short-Term Memory
MHA	Ministry of Home Affairs
NB	Naive Bayes
NER	Named Entity Recognition
NCPC	National Crime Prevention Council
NLP	Natural Language Processing

NLU	Natural Language Understanding
NMF	Non-negative Matrix Factorisation
PCA	Principal Component Analysis
PV-DM	Distributed Memory Model of Paragraph Vector
PV-DBOW	Distributed Bag-of-Words version of Paragraph Vector
RF	Random Forest
RNN	Recurrent Neural Network
SSI	Self-Similarity Index
SDQ	Similarity-Dissimilarity Quotient
SPF	Singapore Police Force
SVM	Support Vector Machine
SMOTE	Synthetic Minority Over-Sampling Technique
TF-IDF	Term Frequency-Inverse Document Frequency
URL	Uniform Resource Locator

Chapter 1

Introduction

1.1 Background

In recent years, scams have become increasingly prevalent alongside greater Internet connectivity and ubiquitous use of mobile devices. With scams predominantly being propagated online, any person around the world can be a potential victim. Scams are as much a cause of concern in Singapore as they are globally. According to official statistics from the Singapore Police Force (SPF), scams constituted almost 17.5% of overall crimes in Singapore in 2018 [1]. This figure rose to 27.0% in 2019 [2]. Between 2017 and 2019, at least S$480 million was cheated from victims by scammers [1–3]. In the first half of 2020 alone, S$82 million was cheated, compared to S$41.6 million in the same period of 2019 [4]. The top four scams of concern in the first half of 2020 were electronic commerce (e-commerce) scams, social media impersonation scams, loan scams and phishing scams [4]. In fact, e-commerce scams had been the most prevalent type of scam in Singapore since 2015 [1–3, 5].

Scams have wide-ranging impacts on victims. The most immediate impact is financial loss, which can in turn cause tremendous emotional stress, particularly if monies lost were significant amounts of savings. Victims also experience embarrassment, shame and humiliation, especially in the case of love scams [6]. Beyond short-term emotional impacts, scams also have longer-term psychological effects on victims, such as increased anxiety and low self-esteem. Scams sometimes have ripple effects on victims' friends and loved ones. In one case, a victim fell prey to a technical support scam and lost more than S$300,000 from joint accounts with family members [7]. Loss of personal data can also be very distressing to victims, especially if identity is misused over a long time.

There have been significant efforts by the SPF to combat scams in Singapore. The Anti-Scam Centre (ASC) was set up in 2019 with a mission to disrupt scam operations [8]. Within a year of its establishment, the ASC froze more than 6,100 bank accounts, recovering at least S$21.2 million, which could have been lost to local and overseas scammers [9]. In addition, the SPF conducts regular police operations against individuals committing scams. In 2019, 85 police operations were conducted against e-commerce scammers, resulting in the arrest of 112

individuals responsible for more than 1,200 cases [2].

To deal with transnational scams, the SPF works closely with foreign law enforcement counterparts. In April 2019, two Nigerian scammers who targeted Singaporean victims in love scams were arrested in Malaysia by the Royal Malaysia Police and extradited to Singapore to face charges [8]. Apart from tough enforcement, public education had also been a priority. The 'Scam Alert' website was launched in 2014 in collaboration with the National Crime Prevention Council (NCPC) with the goal of heightening awareness amongst the public about scams [8]. A more recent example of a public education effort was the launch of the "Spot the Signs. Stop the Crimes." anti-scam campaign in August 2020, which focused on using real-life stories of scam victims as cautionary tales [10].

Despite such efforts, more victims continued to fall prey to scams. This was evidenced by the continuous increase in the number of scam cases reported to the SPF in the last few years. 2019 saw a total of 9,502 police reports relating to scams, almost a two-fold increase from 4,805 in 2017 [2, 3]. More recently, the number of reported scam cases more than doubled in the first half of 2020 compared to the same period in 2019, partly due to more online transactions as Singaporeans stayed home in light of the Coronavirus Disease 2019 (COVID-19) restrictions [4]. One of the biggest challenges in tackling scams is the fact that they transcend geographical boundaries and that a "significant proportion" of scams are being committed by syndicates overseas [2]. Scams also constantly evolve in the ways they target different types of victims and evade detection. The challenge is as much in prevention as it is in enforcement. On one hand, scams exploit victims' vulnerabilities in subtle yet powerful ways. On the other hand, because of these inherent vulnerabilities and the false sense of reality manipulated by scammers, victims believe that they are making reasonable choices [11, 12].

1.2 Research Questions

The gravity of social ills that scams bring about cannot be understated. The psychological impact and economic costs are also significant. There is much more to be done to enhance our understanding of scams and how they can be prevented. This is the main driving force behind our research project. One effective way which we believe would help achieve a better understanding about scams is to draw insights from others' experiences and mistakes. The 'Scam Alert' website was conceptualised precisely for this purpose. It features real-life anecdotes of victims' encounters with scams. These anecdotes, which are in free text, hold tremendous value but are under-utilised. Modern-day machine learning and Natural Language Processing (NLP) techniques can unlock their hidden utility and provide valuable insights towards tackling scams.

The overarching framework of our research is shown in Figure 1.1. The goal of our research was to demonstrate how free text in scam reports can be harnessed using machine learning and NLP methods to develop effective tools for tackling scams. We aimed to apply supervised methods to classify multiple scam types, as well as unsupervised methods to encode scam reports

as document embeddings. Findings from these applications addressed the following research goals, each corresponding to a specific research question:

1. **Finding similar scam reports**: "Given textual description of a scam report, which other scam reports share similar modus operandi?"
2. **Generating key terms**: "Given textual descriptions of a set of similar scam reports, what are the common characteristics of their modus operandi?"
3. **Classifying scam reports**: "Given textual description of a scam report, which category of scam types does it belong to?"

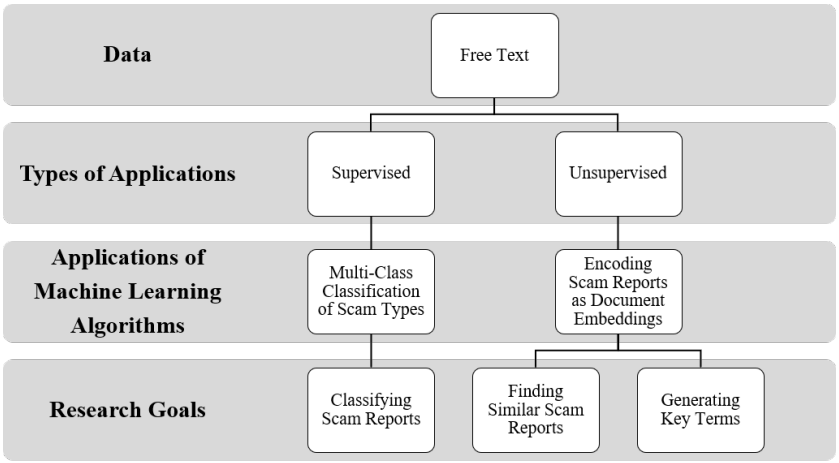


Figure 1.1: Framework of our research

1.3 Outline of Dissertation

The remainder of this dissertation is organised as follows: In Chapter 2, we review academic literature related to the application of machine learning and NLP on free text in the domains of crimes and scams. In Chapter 3, we outline steps taken in preparing our data for modelling and analyses. We explore the data in Chapter 4 to better understand underlying trends. Chapter 5 describes our methodology for supervised multi-class classification of scam reports whereas Chapter 6 does the same for unsupervised encoding of scam reports as document embeddings. In each of Chapters 5 and 6, we begin by first introducing key theoretical concepts, followed by details of our methodology. Thereafter, we present and discuss experimental results before highlighting limitations and alternative approaches which are unique to each area of application. Chapter 7 contextualises our work in Chapters 5 and 6 through a case study of the High Court impersonation scam. In Chapter 8, we discuss the broader significance of our work and how they help towards tackling scams. Finally, in Chapter 9, we summarise key findings from our research and propose different trajectories of machine learning and NLP applications in tackling scams. All analyses underlying our research were performed in Python 3.7.7 [13] and are reproducible using the Python notebooks available at the following Github repository: https://github.com/zeyalt/msc-dissertation-final.

Chapter 2

Literature Review

This chapter presents a literature review, focusing on two themes which are pertinent to our research. The first is the applications of NLP techniques on free text in domains of crimes and scams. Literature around these applications were categorised according to various NLP tasks. The second theme pertains to a theoretical understanding about what scams entail and what makes scams successful. Ultimately, the review of literature around both themes helped in identifying knowledge gaps, which in turn motivates our work in using NLP techniques on free text to tackle scams.

2.1 Natural Language Processing for Crimes and Scams

The application of NLP and machine learning methods on unstructured free text in the domains of crimes and scams is not new. This section highlights related research in four main NLP tasks: named entity recognition, natural language understanding, topic modelling and classification.

2.1.1 Named Entity Recognition

Named Entity Recognition (NER) is an NLP task that picks out salient entities such as names and locations from a given text [14]. One of the earlier works applying NER in the crime domain was by Chau, Xu and Chen in 2002. In their work, named entity extraction techniques were used with a neural network to extract entities such as names, addresses and drug names from police reports in free text [15]. Named entity extraction later evolved into NER as we know it today.

Several more recent studies that used NER had sought to maximise the utility of entities extracted. For instance, the works of both Al-Zaidy *et al.* [16] and Elyezjy and Elhaless [17] went beyond simply extracting entities to constructing meaningful crime networks using those entities. These networks could then be analysed to support crime investigations and identify criminal links. Another way in which entities extracted using NER could be further tapped on was proposed by Schraagen *et al.* [18]. In this collaboration work between the Dutch National Police and Utrecht University, information extracted from crime reports were used in a formal

reasoning system. This system enabled automatic formulation of questions, which could be posed to complainants to clarify on their reports. This enhanced the quality of reports as well as the efficiency in the processing of reports by the Dutch National Police.

2.1.2 Natural Language Understanding

Natural language understanding is the ability of machines to understand the semantics of a given text in natural language [14]. The Dutch National Police and Utrecht University had, in fact, collaborated on several projects leveraging free text. These collaboration appeared to be steered towards creating intelligent agents capable of having natural dialogues with human complainants of crimes. This entailed creating a knowledge graph from a crime report, which could be used to reason about the described incident and if necessary, formulate follow-up questions to be clarified with the complainant [18–20].

Based on the existing body of related literature, these work by the Dutch National Police and Utrecht University stand out as one of the recent state-of-the-art developments in terms of revolutionising the ways and the extent to which unstructured free text could be used in the crime domain. To our knowledge, no other research had ventured into the realm of creating agents intelligent enough to converse with human complainants about their reports. Their work also represented a significant stride towards machines attaining natural language understanding, currently a major challenge in Artificial Intelligence (AI).

2.1.3 Topic Modelling

Topic modelling is the application of statistical machine learning techniques to identify key topics relevant to a set of documents [21]. One unsupervised topic modelling technique is known as Non-negative Matrix Factorisation (NMF). It uses linear algebra to decompose high-dimensional text data into lower dimensions, from which underlying topics can be derived [22].

In [23], Kuang, Brantingham, and Bertozzi used NMF to discover latent topics from crime reports in free text. The team referred to this approach as "soft-clustering" of crimes, which enabled topics to "emerge autonomously" from a set of crime documents [23]. The team asserted that these topics captured "behavioural and situational conditions" of crimes and that such information would otherwise not be found if crimes were classified into distinct categories. One similarity between their work and ours is the idea that such free text data inherently contains behavioural and situational information. In our case, textual descriptions in scam reports held useful information about victims' and scammers' behaviours, as well as situational factors involved in scams. The difference is that while such information was captured as latent crime topics in [23], they did so in the form of key words and phrases in our work.

Another work that applied topic modelling on police narrative free text was that by Birks *et al.* [24]. In their work, Birks *et al.* used Latent Dirichlet Allocation (LDA), which is a probabilistic algorithm that assumes each document to contain a mix of different latent topics [25].

Birks *et al.* demonstrated how LDA was used to identify latent topics from free text describing residential burglaries. Each document describing a single burglary incident was assigned a dominant topic. Thereafter, documents were grouped by dominant LDA topics into meaningful clusters, where each cluster related to similar modus operandi. The works of both Kuang *et al.* [23] and Birks *et al.* [24] resonated strongly with our research because of the common motivation to analyse modus operandi by proactively leveraging free text data using NLP methods.

2.1.4 Classification

Classification was arguably the most ubiquitous NLP task in the crime domain. Based on current literature, there was a significant amount of research in the detection of cyber-crimes, in particular, phishing scams. In these studies, detection of phishing scams was modelled as a binary classification problem, where the possible outcomes were either phishing or non-phishing. Notably, a variety of data sources were used to detect phishing. These include electronic mail (e-mail), website, social media profile and Uniform Resource Locator (URL).

One of the earlier studies that used machine learning and NLP methods to detect phishing was by Abu-nimeh *et al.* in 2007. Abu-nimeh *et al.* used a training dataset comprising both phishing and non-phishing e-mails. They compared several machine learning classifiers, including Logistic Regression (LR), Support Vector Machine (SVM) and Random Forest (RF), and found that RF outperformed the rest in terms of predictive accuracy [26]. In retrospect, their work, being one of the firsts in using machine learning and NLP to detect phishing, was likely to be instrumental in laying the foundation for subsequent research. There were more than 250 academic citations of their work, one of them by Mbaziira and Jones [27].

Mbaziira and Jones similarly used e-mails to detect phishing scams, but with SVM, Naive Bayes (NB) and *k*-Nearest Neighbours (KNN) classifiers [27]. Mbaziira and Jones also experimented with textual data collected from Facebook profiles linked to known Nigerian cybercriminals and found that the NB classifier produced best accuracy in predicting fraudulent profiles [27]. In a separate work, Mbaziira used bilingual text datasets in English and Nigeria Pidgin and found SVM to be superior over NB and KNN in detecting phishing scams [28].

In several research papers, URLs were used to detect a phishing site. The general approach was to analyse URLs and websites of both malicious and non-malicious content, and to extract features as inputs into machine learning algorithms. This approach was taken in [29], where features of URLs and websites were fed into an artificial neural network to classify websites as phishing or non-phishing.

The architectures of neural networks used in [29] were shallow, consisting of at most two layers and eight neurons. Since then, other studies had used deep neural networks instead. An example was [30], where recurrent neural network was used to predict phishing sites. Another example was the work of Chatterjee and Namin [31], which implemented deep reinforcement learning to detect malicious URLs. Their work was ground-breaking because of the ability of

their model to adapt to the dynamic behaviours of phishing websites and automatically learn features to detect phishing [31].

In addition to using supervised learning, where training datasets were labelled, unsupervised algorithms had also been used to detect phishing scams. Feng *et al.* [32] treated the Document Object Model (DOM)[1] of websites as natural language. They used a document-to-vector (doc2vec) model to convert DOM of websites into vector representations, before clustering similar websites using Manhattan distance between vectors. This work is of particular relevance to our research. As we will see in Chapter 6, doc2vec was used to represent scam reports as vectors. The difference between their work and ours is that we had used cosine and Jaccard similarities instead of Manhattan distance to find similar scam reports. Another clever application of unsupervised learning in this area was by Wu *et al.* [33], where crypto-currency transaction records were represented as vectors using a novel algorithm called *trans2vec* before being classified as phishing or non-phishing using a SVM classifier.

2.2 Understanding Scams

Besides understanding how machine learning and NLP methods were applied in the domains of crimes and scams, we also examined studies relating to scams from theoretical perspectives. We envisioned these studies to not only help in shaping the definition of scams in our work, but also enable a better appreciation of what makes scams successful.

2.2.1 What Is a Scam?

To date, there is no consistent definition of "scam" in current literature. One general definition was "a fraudulent scheme involving money and some sort of business transaction" [34]. Another emphasised the role of persuasive language and defined a scam as a fraud where money is extorted by "manipulating language to distort reality" [11]. The latter definition focused on the role of language but was ambiguous on how "reality" is distorted. Both definitions were inadequate in encompassing scams which involved loss of personal information. Pouryousefi and Frooman, on the other hand, attempted to define scams by the key characteristics of individual scam types [35].

Understandably, defining scams in a consistent manner is difficult given their complex nature and how quickly they evolve. Notwithstanding, it is also imperative to be explicit about the context in which scams would be discussed in this dissertation. Taking into account current literature as well as Singapore's context, we defined a scam as "a scheme that is designed to deceive individuals into giving away their money or personal information, generally by using the Internet". It is this definition that subsequent discussions in the remainder of the dissertation

[1]The Document Object Model is an application programming interface for valid hypertext markup language (HTML) and well-formed extensible markup language (XML) document such as web pages. It defines the structure, style and content of documents.

will be premised on.

In existing literature, it was not uncommon to find the word "scam" being used synonymously with "fraud". It is necessary to clarify that our research treated the two terms as different. We posit that the main distinction is whether a scheme successfully causes the victim to lose money or personal information. Within Singapore's legal framework, a fraud is tantamount to a criminal offence of cheating, a necessary element of which is the "delivery of any property[2] to any person" by the person being deceived, as a result of the deception [36]. Based on our definition of a scam above, it is not necessary for a property to be completely delivered. A scheme can be considered a scam, regardless of whether the victim suffers any losses, as long as it entails an intention to deceive. In other words, a scam can be a failed attempt at fraud, as was often the case from victims' accounts published on the 'Scam Alert' website.

2.2.2 What Makes Scams Successful?

Having made explicit the definition of scams in our research, we next review literature on what makes scams successful. This discussion is based on the Routine Activity Theory, which is a sociological theory proposed by Cohen and Felson in 1979 [37]. It states that the following three elements must converge in time and space for a crime to occur: suitable target, motivated offender and an absence of a capable guardian [37]. We believe that this theory is applicable in the context of scams, though the three elements need not converge in time and space as in physical crimes. Scammers are offenders who are motivated to use different ways to deceive individuals. A capable guardian can be any person who intervenes and prevents another person from becoming a scam victim. Victims also often make their weaknesses known, such as yearning for companionship and desire for cheap deals, making themselves suitable targets to scammers. Without a capable guardian, individuals become suitable targets when they encounter a motivated offender.

Much research had been conducted to study attributes of victims that make them susceptible to scams. Amongst these attributes are low self-control [38, 39], impulsiveness [40], perception towards the size of reward [41] as well as loneliness [6]. There were mixed findings about how a person's savviness with technology and the Internet influences his susceptibility to scams. While Wright and Marett found that being more Internet-savvy led to lower susceptibility [42], Wilsem [38] observed that Internet use per se did not protect victims regardless of their savviness. In his study, Wilsem [38] reported that there was an increased likelihood of victims responding to fraudulent messages, particularly those dealing with higher volumes of e-mails.

Given that scams typically require planning and resources to execute, we postulate that the Rational Choice Theory applies to scammers. This means that scammers are rational thinkers, making rational choices when committing scams [43]. These choices include actions taken to increase chances of success. For instance, scammers invoke visceral appeals from victims,

[2]Legally, "property" can include money, virtual money and personal information [36].

including appeals to love, in the case of love scams [44], and authority, in the case of scams impersonating government officials [45]. They expend effort to design messages that mirror official communications or require an urgent response in order to increase rates of response from victims [44, 46, 47]. In addition, their skillful and persuasive use of language make victims feel like they are making sound decisions within a false reality they created [11].

Lack of capable guardians is another factor determining the success of scams. In their study, Graham and Triplett used digital literacy as a measure of guardianship and discovered that respondents with higher digital literacy reported receiving phishing emails more [48]. Their study suggested that higher levels of guardianship meant greater awareness of phishing. Another study found that locations which were disadvantaged in terms of receiving "adequate levels of capable guardianship" were associated with higher fraud risks [49]. In light of these findings, one of the ways to curb scams is to nurture capable guardians within the society. It is with this motivation that our research involved generating key words and phrases from similar scam reports. These key words and phrases can potentially help law enforcement authorities identify specific touch points, such as banks and money remittance companies, who can play vital roles as guardians in the society against scams.

2.3 Summary

We have reviewed related work from both methodological and theoretical perspectives. In Section 2.1, we examined past research that applied machine learning and NLP methods on free text in the domains of crimes and scams. Most research that were related to scams appeared to focus specifically on phishing scams. Moreover, they were largely limited to the binary classification task of detecting phishing scams. On the other hand, the theoretical discussions in Section 2.2 set the context of how scams were defined in our work, as well as provided a grounding for understanding why scams succeed based on the Routine Activity Theory.

Besides the 'why', it is also of interest to understand the 'how' — how scams happen. Indeed, current literature appeared to be limited in terms of discovering modus operandi of scams. One way to bridge this gap is script analysis. In the context of crimes, script analysis is a step-by-step breakdown of the events involved in a crime [50]. From the offender's perspective, it encompasses actions taken by a criminal before, during and after the crime process [50]. To our knowledge, no previous studies had used script analysis to analyse cyber-crimes, let alone scams. We envision that a script analysis of scams can provide an organised framework for studying their modus operandi. Another potential value of scam script analysis is that it can help us identify points of intervention where scams can be disrupted. As we will see in Chapter 7, our work in generating key words and phrases from similar scam reports can aid greatly in the process of developing scam scripts.

Chapter 3

Data Preparation

Having gained an appreciation of related literature, we now turn the focus on our research, which we begin by preparing our data. This chapter lays out the procedures taken to prepare our data for subsequent analyses and modelling. We first provide a background about the data source and define key terminology. This is followed by step-by-step explanations about how our text data was extracted, cleaned, feature-engineered and pre-processed.

3.1 Data Source

The data used in our research was derived from the 'Scam Alert' website[1]. 'Scam Alert' was launched in 2014 by the NCPC. The NCPC is a non-profit organisation in Singapore that is actively involved in educating the public about crime through crime prevention campaigns, exhibitions and talks. The NCPC has been playing an instrumental role in the fight against scams in Singapore. 'Scam Alert' is one of its many initiatives aimed at promoting awareness about scams and providing avenues of help to victims.

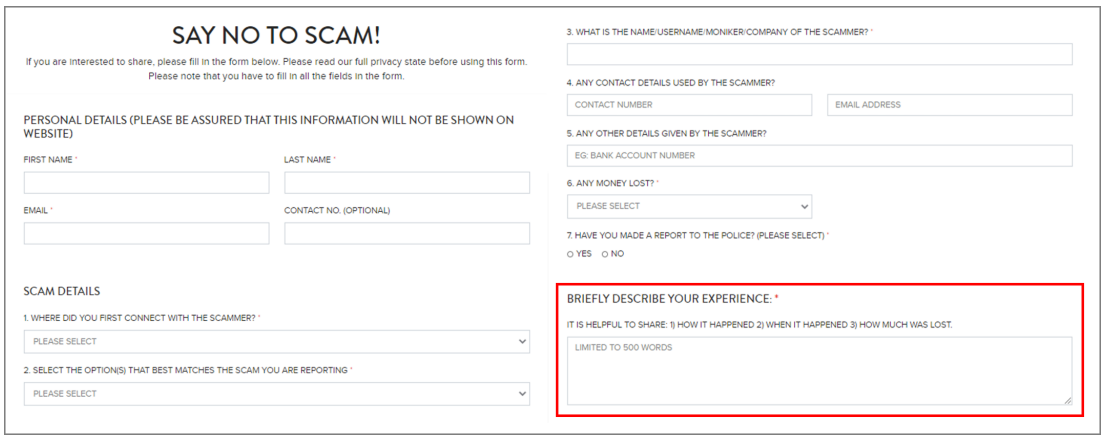


Figure 3.1: A screenshot of the 'Share a Story' form on 'Scam Alert'

[1]The 'Scam Alert' website is accessible at the following URL: https://www.scamalert.sg/.

'Scam Alert' contains resources about different types of scams in Singapore as well as tips to avoid becoming a victim. Members of public can share their encounters with scams on the website, in which case they will be directed to fill in a form, as shown in Figure 3.1. Victims can describe their scam experiences using the free text field, shown by the red box in Figure 3.1. These textual descriptions form the crux of our research.

Once a scam story is submitted, it will first be vetted by NCPC. The vetting process includes omitting personal or sensitive information as well as removing submissions which do not pertain to scams, such as civil disputes. When filling the form, victims are required to select, from a drop-down menu, a scam category that best matches their scam stories. The selected categories are also reviewed by NCPC during the vetting. A scam report will be reclassified if its original category do not match the scam story. After the scam reports have been vetted, they will be published on the website within 24 hours, or the next working day if they were submitted on weekends or public holidays [51].

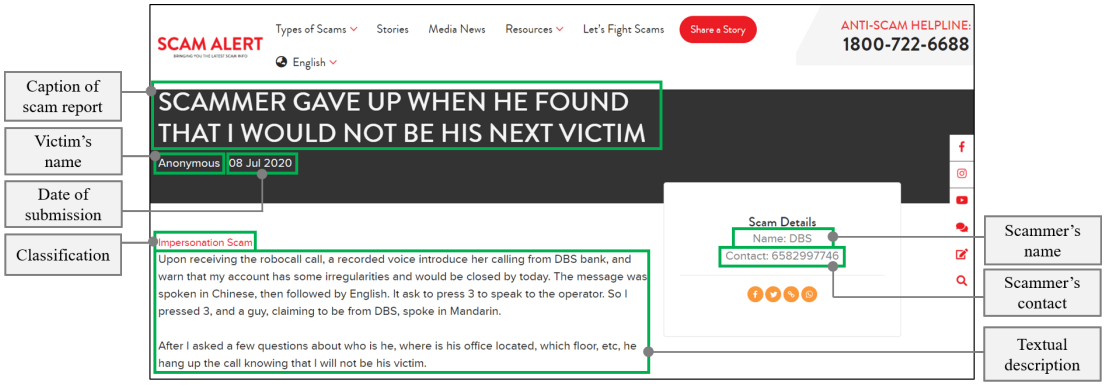


Figure 3.2: A screenshot of a scam report published on 'Scam Alert'

Figure 3.2 shows a screenshot of a published scam report containing various information. Amongst them, the most relevant and important for our research were the textual description and the classification. Unlike official police reports, these scam reports do not receive police attention. Victims submitting scam reports may choose to lodge police reports separately. Conversely, those who lodge police reports on scams may not necessarily share their experiences on 'Scam Alert'. This non-mutual exclusivity means it is expected that there will be inconsistencies between trends in the data from 'Scam Alert' vis-à-vis official scam cases reported to the SPF. This will be further explored in Section 4.1.2.

3.2 Key Terminology

The following is a list of key terminology which will be used in this dissertation. As far as possible, they are aligned with the standard terminology in NLP.

- *Text corpus* is the collection of scam reports containing textual descriptions of victims' encounters with scams.

- A *scam report* refers to a single submission of a scam story on 'Scam Alert'. It contains textual descriptions of victims' encounters with scams. It is synonymous with *report* and *document*.
- A *token* is the basic building block of the text in a scam report. It refers to a word, a punctuation mark or a set of characters such as emojis.
- A *victim* is any person who shares his or her scam experience by submitting a scam report on 'Scam Alert', without necessarily having fallen prey.
- A *scam type* refers to one of the defined categories of scams. It is synonymous with *scam category*, *class* and *classification*.

3.3 Data Extraction and Cleaning

These scam reports published on 'Scam Alert' are a repertoire of free text data which hold tremendous value and insights about scams in Singapore. To access and use these scam reports for our research, we first obtained written permission from the data owner, NCPC [51]. The scam reports were extracted from 'Scam Alert' using web-scrapping techniques with the Python libraries, `BeautifulSoup` [52] and `Selenium` [53].

There were two stages involved in extracting the scam reports. Since each web page displayed six scam reports and each scam report had a unique URL, we first extracted the six URLs from a single web page and then repeated the process for all 800 web pages. This generated 4,796 URLs. The second stage was processing each URL to extract the contents of each scam report. The extracted information included date of submission, textual description, scammers' details and scam type. The extracted data spanned four years, between 20 July 2016, when this reporting feature first started, and 19 July 2020. Excluding 136 URLs which were found to be invalid, a total of 4,660 scam reports were extracted.

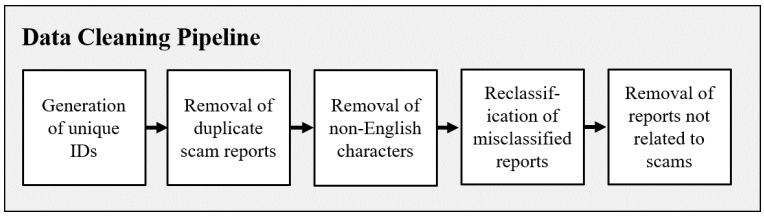


Figure 3.3: Data cleaning pipeline

The next step was to clean the data. Our data cleaning pipeline is summarised in Figure 3.3. To begin the process, we generated a unique alpha-numeric identification (ID) code for each scam report. This was to enable unique identification of scam reports for downstream tasks such as reclassification of misclassified reports. Next, we removed 64 duplicate scam reports which were assessed to contribute limited information. Following this, we checked for reports containing non-English characters and found 96 that contained Chinese characters. Of

these 96 reports, six were entirely in Chinese characters. One option was to translate them into English. However, since they constituted less than 1% of all scam reports in our text corpus, it was assessed that removing them would not significantly affect our analyses. For the remaining 90 reports with Chinese characters, the characters occupied only a small segment of each text. Hence, only the characters were removed, retaining bulk of the text in English.

An important step in cleaning our data was to ensure that scam reports were accurately classified. To do this, we used two methods. First, we examined reports with extremely short textual descriptions. The assumption was that unusually short descriptions would unlikely be related to scams or contain meaningful information. Second, we manually inspected textual descriptions of reports in each category. Given that it was a laborious task to inspect every scam report, we adopted a tiered approach. For scam types with 100 reports or less, we inspected every report; for scam types with at least 100 reports, we inspected a random sample of 20%.

Using this approach, a total of 1,232 scam reports were manually inspected. This constituted about 27% of the entire text corpus. Out of these 1,232 scam reports, 292 were found to be either misclassified or unrelated to scams. A scam report was considered to be misclassified if its text did not match the scam type, according to the descriptions of various scam types on 'Scam Alert'. These descriptions are presented in Appendix A. Table 3.1 shows selected examples of misclassified reports as well as their original and amended classifications.

Table 3.1: Examples of misclassified scam reports

ID	Text	Original Classification	Amended Classification
20200504-0D0m3t	I received a call from a guy with a thick indian accent. he said my wife full name and WANT to SPEAK to her. He said he is from microsoft support claiming my wife computer had security issues, he need to install driver for her. Knowing that things doesn't add up. I hung up the phone.	Software Update Scam	Impersonation Scam
20180830-s6RlkJ	one thousand seven hundreds singapore dollars	Impersonation Scam	Not relating to scams
20180822-xAr3gu	Automated robot voice called me 63460644 and said it is from Singapore High court, and I have outstanding summon??? Also repeated in mandarin then I hung up. Not sure what is the intention of the call.	Phishing Scam	Impersonation Scam
20161120-xMzghF	I was reading this article about how this homeless man bought a Ferrari by getting money on this website. I tried the website and it asked for my particulars. I put down my email and clicked send but only then did I realize it was a scam. For a while a man with a European accent called me and sent me emails saying he could make me a million dollars. I have recorded the calls	Money Mule Scam	Phishing Scam

Out of the 292 misclassified reports, 36 were unrelated to scams and were removed from our

corpus. The vast majority of the misclassified reports were reclassified as impersonation scams. This could be due to the relatively broad definition of impersonation scams. For example, many reports which were initially classified as software update scam involved perpetrators impersonating as staff from companies such as Microsoft. These reports also fitted the description of an impersonation scam and were therefore reclassified accordingly. The remaining 4,554 scam reports formed the text corpus for our research.

3.4 Feature Engineering

Having cleaned the data, the next step in data preparation was feature engineering. Feature engineering entails wrangling the data to extract or create new features, which can then be fed into machine learning algorithms as inputs. In our case, feature engineering was an essential step given the highly unstructured nature of our raw dataset. It allowed us to better understand the underlying trends in our data. Figure 3.4 summarises our feature engineering workflow.

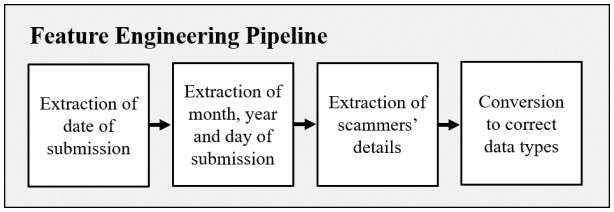


Figure 3.4: Feature engineering pipeline

Table 3.2: Selected scam reports from raw corpus, before feature-engineering

ID	Submission Details	Scam Type	Incident Description	Scammer Details
2020 0717-yOxIA1	Anonymous — 17 Jul 2020	Phishing Scam	it happened this morning 0913hrs i received a phone call from ... Knowing that it was a scam, i hung up the call.	Name: Ministry of law Contact: +6566309459
2020 0715-JnHFae	Anonymous — 15 Jul 2020	Impersonation Scam	on 15 Jul 2020 at 9.51am, received a phone call from ... I have a parcel not collected... immediately I hang up the call!	Name: Ministry of health Contact: +6568508182

Table 3.2 shows two scam reports from our raw text corpus. The first feature engineering step was to extract dates of submissions from the 'Submission Details' column and store them in separate columns. Several new features were extracted from dates of submissions. They include month, year and day of the week of submissions. These features enabled subsequent temporal analysis of the data. Similarly, scammers' names and contact numbers were extracted from the 'Scammer Details' column. Table 3.3 shows the same scam reports of Table 3.2 after feature

engineering.

Table 3.3: Selected scam reports after feature engineering

ID	Year	Month	Date	Day	Scam Type	Incident Description	Scammer Name	Scammer Contact
2020 0717-yOxIA1	2020	07	17	Fri	Phishing Scam	it happened this morning 0913hrs i received a phone call from ... Knowing that it was a scam, i hung up the call.	Ministry of law	+656630 9459
2020 0715-JnHFae	2020	07	15	Wed	Imperson-ation Scam	on 15 Jul 2020 at 9.51am, received a phone call from ... I have a parcel not collected... immediately I hang up the call!	Ministry of health	+656850 8182

Another key step in feature engineering was ensuring that every feature were of the correct data type. We converted the 'Date' column to a date-time object, a data type in Python that is suitable for handling dates and times. We also converted the 'Scam Type' column, as well as columns containing month, year and day of submissions to categorical data types. This facilitated visualisation of trends during our exploratory data analysis.

3.5 Text Pre-Processing

Text, in its raw form, is highly unstructured and noisy, making it difficult for machines to interpret and understand. Text pre-processing is therefore an essential step in any NLP task which helps to make the text corpus more consistent. This current section elaborates on the text pre-processing pipeline, which had been designed to clean our text corpus for specific purposes of our work in this research. The text pre-processing pipeline is shown in Figure 3.5.

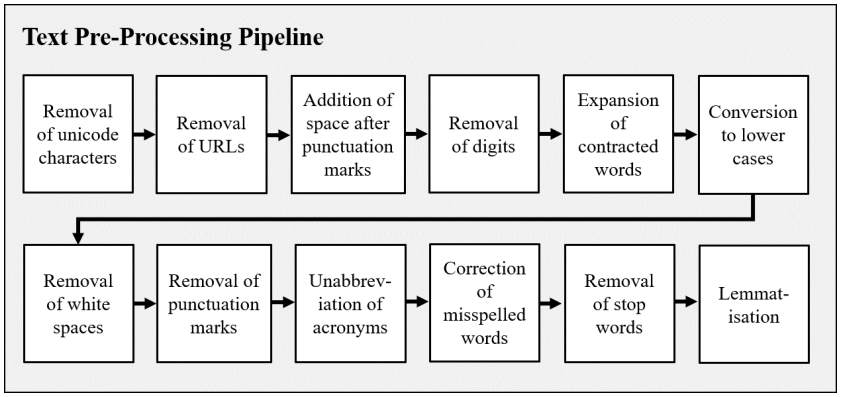


Figure 3.5: Text pre-processing pipeline

The first step involved removing unicode characters from the text. Though not prevalent, the text corpus contained several unicode characters like `w\u200cer\u200ce`, `n\u200cext` and `i\ufeffmmediately`. Upon closer inspection, it was found that these characters were embedded within words. For instance, `n\u200cext` was the characters `\u200c` embedded within the word `next`. Given that such words would be of utility to our corpus, we preserved the words by removing the unicode characters only.

Next, we removed URLs and replaced them with a place-holder, `url_link`. Due to huge variations in the styles of writing by victims, it was not uncommon to have instances where there was no space between words and punctuation marks. Consider the following example: "`Got call from a lady,with Indian accent...`". Without a space between `lady` and `with`, subsequent steps to remove punctuation marks and tokenise would inadvertently combine them as a single token, `ladywith`, thus potentially interfering with good model performance. To resolve this, in instances where there were no spaces between words and punctuation marks, we used regular expressions to add a space between them.

As far as our research goals were concerned, digits such as date, time and telephone numbers were assessed to be unnecessary. They were hence removed. Having said that, such numerical information would otherwise be of importance in other NLP tasks such as NER. The next pre-processing step involved expanding contracted words, which is a common practice in NLP to standardise text. For instance, "don't" was expanded to "do not" and "can't" to "cannot". Other steps which helped to standardise text in our corpus were converting words into their lower cases as well as removing white spaces and punctuation marks.

The next step was unabbreviation of acronyms. The general approach to find acronyms was using regular expressions to detect consecutive upper-case letters, such as "IBAN" representing the phrase "International Bank Account Number". Acronyms which were written in lower cases were identified manually. The list of 63 acronyms we found is presented in Appendix B. These acronyms were saved as a Python dictionary, with the acronyms as keys and the unabbreviated forms as values. Using this dictionary, acronyms in our text corpus were replaced by their unabbreviated forms.

Given that our text corpus originated from victims with different levels of language proficiency, there were huge linguistic variations. One example is in the spellings of words. There were plenty of typographical errors in our corpus. Rectifying them was another crucial step in text pre-processing and arguably the most laborious. Unlike an acronym which typically corresponded to a unique phrase, words could be misspelled in several different ways. For instance, there were at least 13 different spellings of "WhatsApp", the popular text messaging mobile application.

A two-pronged approach was taken to rectify typographical errors. The first involved using `pyspellchecker`, a Python library that provides a spell-checking algorithm. In essence, `pyspellchecker` aided us in identifying misspelled words in the text corpus using the Lev-

enshtein distance algorithm[2]. Its limitation, however, was that its effectiveness in spell-checking depended on the known words. In our case, it failed to detect typographical errors of lesser known words or words unique to our text corpus. For instance, "carousell" and "shopee" are names of two online shopping platforms in Singapore. Misspellings of these words like "carousel" and "shoppe" were undetected using `pyspellchecker`. To mitigate this, we adopted the second approach, which was to manually identify misspelled words as and when they were spotted during our research. A non-exhaustive list of 100 misspelled words is shown in Appendix C. Notwithstanding, we recognise that it would be practically infeasible to correct all typographical errors. Hence, the premise of this two-pronged approach was to identify misspelled words which, if corrected, could help further standardise the text and contribute meaningful information.

After spell-checking, the next step was to remove stop words, another common text pre-processing step in NLP. Stop words are words that occur frequently in any text. They play important roles in grammar but contribute limited meaning to the text [14]. Examples are "to", "an", "the", "which" and "it". Removing stop words allows machine learning algorithms to focus on words which better define meanings of text. To perform this task, we used the `NLTK` library [55], which had a built-in list of 179 stop words of the English language. Removing stop words reduced the total number of words in our text corpus by almost half.

The last step in text pre-processing was lemmatisation. Words vary in terms of morphology or structure. The basic structure of words consists of prefixes such as "de-", "un-" and "auto-", suffixes such as "-ing", "-ed" and "-ly", as well as the root words. Words can share the same root but have different prefixes or suffixes. For example, the words "play", "plays", "played" and "playing" have different grammatical meanings but are derived from the same root word, "play". Before any NLP tasks, it is prudent to reduce morphological variations of words [14]. There are two methods to do this. The first is *lemmatisation*, which reduces words to their root words [14]. These root words are valid words in the English dictionary. The other method is *stemming*, which refers to slicing the part of a word that varies morphologically [14]. Table 3.4 shows examples of raw words, after lemmatisation and stemming respectively.

Table 3.4: Effects of lemmatisation and stemming on raw words

Raw Word	Lemmatisation	Stemming	Raw Word	Lemmatisation	Stemming
Buy	Buy	Buy	Receive	Receive	Receiv
Buys	Buy	Buy	Receives	Receive	Receiv
Buying	Buy	Buy	Receiving	Receiving	Receiv
Bought	Buy	Bought	Received	Receive	Receiv

There are advantages and disadvantages to lemmatisation and stemming. Firstly, lemmatisation reduces the number of unique tokens in the text corpus, which translates to lesser noise

[2]In the Levenshtein distance algorithm, permutations of words within an edit distance of two were compared against known words [54].

[56]. Secondly, lemmatisation considers the context of a word using its part-of-speech tag, such as a verb or a noun [57]. These advantages allow algorithms to focus on root words and learn meanings of text better. On the other hand, stemming is computationally more efficient than lemmatisation [57]. One downside to stemming, however, is that it does not necessarily produce the same root word. As we see in Table 3.4, stemming converted the words "buying" and "bought" to "buy" and "bought", even though they were from the same root word "buy". Stemming also generated invalid words like "receiv". Given these considerations, it was assessed that lemmatisation would be more effective for subsequent applications of machine learning and NLP methods in our research.

3.6 Summary

A methodical process of data preparation is pivotal in yielding better results in any downstream machine learning tasks. This was particularly true for our text data, which was highly unstructured and contained a great degree of linguistic inconsistencies. In this chapter, we have seen how the text data was first extracted from 'Scam Alert', before being cleaned, feature-engineered and pre-processed. Although our text data is now in a format suitable for modelling with machine learning algorithms, we will, in the next chapter, first explore the data in order to gain a better understanding of underlying trends and characteristics.

Chapter 4

Exploratory Data Analysis

This chapter provides a deeper understanding of our data through Exploratory Data Analysis (EDA). It uses the text data that has been fully pre-processed from the previous chapter. There were three key benefits to performing EDA. First, EDA revealed underlying trends in our data, such as the usage of 'Scam Alert' as a platform for victims to share their scam experiences. Second, it allowed us to verify assumptions about our data. For instance, comparing our data against official statistics enabled us to understand how representative our data was of the actual trends of scams in Singapore. Third, EDA of the textual information gave us useful insights on the most common and important terms for each type of scam. The first part of this chapter presents results of EDA from temporal and non-temporal analysis, as well as the common types of scams being reported on. The latter part examines textual information in scam reports, identifying words which are common and unique to each scam type.

4.1 Exploratory Analysis of Scam Reports

4.1.1 Temporal Analysis

Given that submissions of scam reports on 'Scam Alert' by victims were voluntary, statistics pertaining to these submissions might not accurately represent the underlying trends of scams in Singapore. They could, however, provide an understanding of the usage of this platform by victims to share their scam stories. This section explores this through a temporal analysis.

Figure 4.1 shows a time series graph, with a 14-days moving average, indicating how submissions of scam reports had varied with time. Since this platform started in July 2016, the number of scam reports being submitted increased steadily until mid-2018, when there was a sharp increase. This was followed by a drop towards the end of 2018. The first half of 2019 saw a very low usage of this platform. Thereafter, its usage increased again, peaking again sometime around the first quarter of 2020.

Although our data might not accurately reflect actual trends, they allowed us to understand

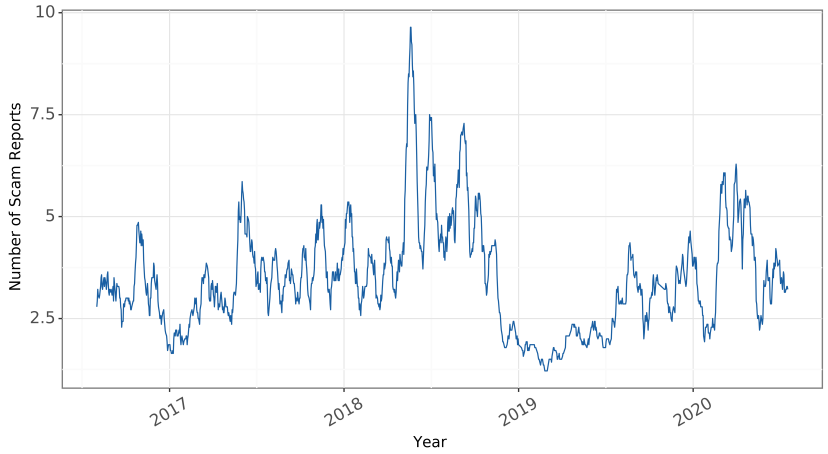


Figure 4.1: Time series visualisation of scam report submissions on 'Scam Alert'

the relative prevalence of different types of scams in Singapore over time. Figure 4.2 shows time series graphs of scam report submissions, with a 14-days moving average, for the top 12 scam types. Several key insights can be drawn. First, impersonation scams and to a lesser extent, internet love scams, appeared to have been consistently prevalent in the last four years. Second, several scam types seemed to have declined in prevalence since the start of 2019, evidenced by the flattening of the respective time series graphs. These include online purchase scam, cyber extortion scam and investment scam. Third, trends for scams like loan and phishing scams, on the other hand, suggested growing prevalence.

In Figure 4.1, we observed an increase in submissions of scam reports in the first half of 2020. Moreover, there was also an increasing trend of impersonation scams in Figure 4.2 for the same period. Further analysis was performed to investigate if this phenomenon was contributed by the COVID-19 pandemic. Using `spaCy`, an open-source Python library for NLP tasks [58], we extracted scam reports containing words and phrases related to the COVID-19 situation in Singapore's context. Examples include "pandemic", "masks", "hand sanitizers" and "circuit breaker"[1]. Out of 702 scam reports in 2020, 110 were found to contain such words and phrases. Only 70 out of these 110 scam reports pertained to scams exploiting the COVID-19 situation. These 70 reports were submitted between 13 February 2020 and 15 July 2020, with peaks sometime around 27 March 2020 and 17 April 2020.

62 out of these 70 reports were classified as impersonation scams. Closer inspection revealed that they involved perpetrators impersonating as Ministry of Health officials in order to solicit victims' personal information. These preliminary findings provided some evidence that the increase in the number of submitted scam reports and impersonation scams in the first half of 2020 was attributable to scams taking advantage of the COVID-19 situation. According to the Ministry of Home Affairs (MHA), besides impersonation scams, other scams connected to

[1] "Circuit breaker" was a term used to describe the measures implemented by the Singapore Government to contain the spread of COVID-19.

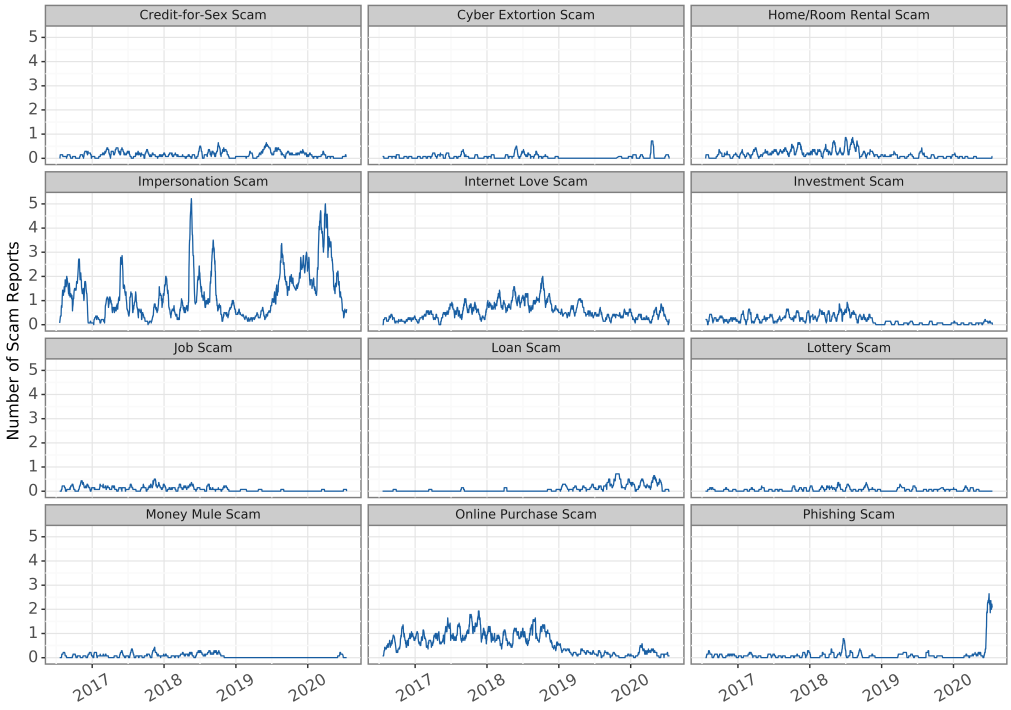


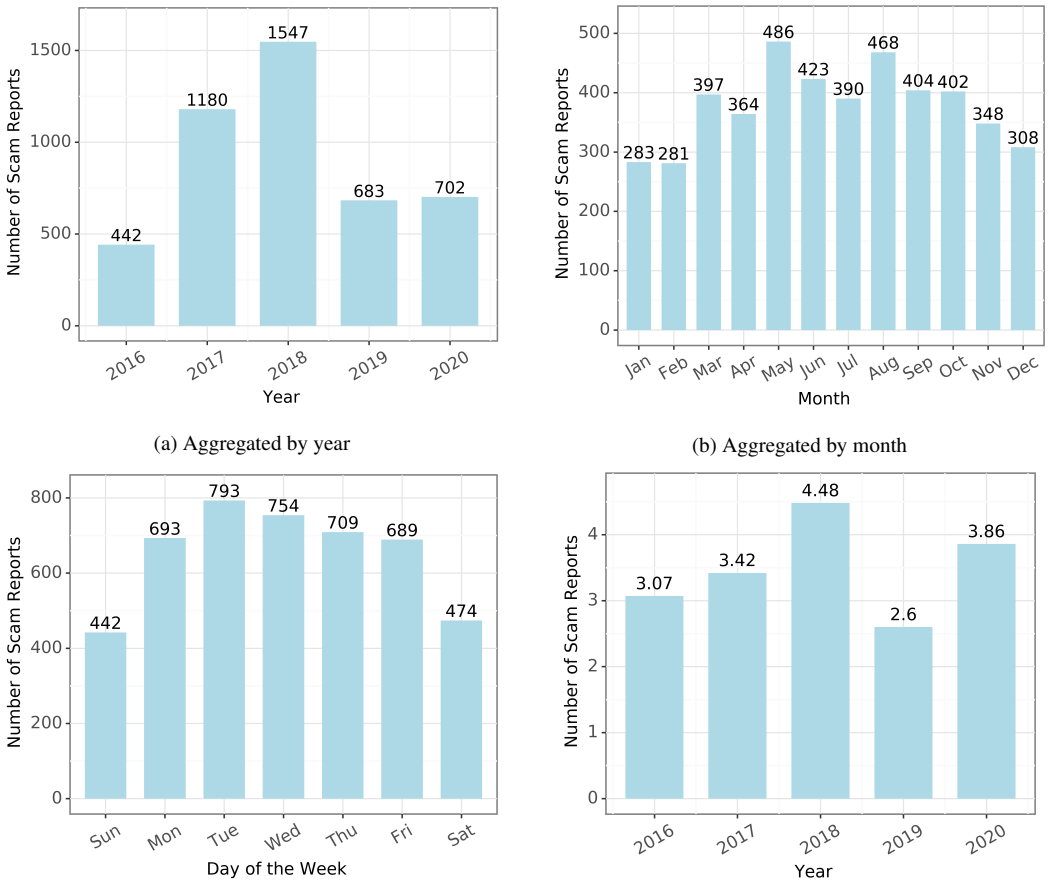
Figure 4.2: Time series visualisations of scam report submissions for top 12 scam types

COVID-19 included online purchase and phishing scams [59]. However, there were only four reports each for online purchase and phishing scams amongst the 70 that we found. It was therefore unlikely for these eight reports to have contributed significantly to the increasing trend in Figure 4.1 for the first half of 2020.

4.1.2 Statistical Summaries

This section presents overall statistical summaries of the 4,554 scam reports, aggregated by year, month, day of the week and daily average. These are presented in Figure 4.3. From Figures 4.3a and 4.3d, it is evident that 2018 recorded the highest absolute and daily average numbers of scam reports submitted on 'Scam Alert'. 2018 saw a total of 1,547 submitted scam reports, with a daily average of 4.48. This was consistent with the observation from Figure 4.1 of the increase in submissions in mid-2018. Conversely, 2019 had the lowest daily average submissions.

Based on official statistics from the SPF, the total number of reported scam cases increased from 4,805 in 2017, to 6,189 in 2018 and then to 9,502 in 2019 [1, 2]. This implies that for every victim who submitted a scam report on 'Scam Alert' in 2017, there were four who lodged police reports on scams. This ratio stayed roughly the same in 2018 but increased more than three-folds in 2019. This suggested that 2019 was an unusual year in terms of scam report submissions on 'Scam Alert'. Figure 4.3b indicates that the months of May and August saw the most submissions, whereas the months of January and February saw the least. From Figure 4.3c, we observe that an average of 728 scam reports were submitted on weekdays, substantially



(a) Aggregated by year

(b) Aggregated by month

(c) Aggregated by day of the week

(d) Daily average by year

Figure 4.3: Number of scam reports by year, month, day and daily average

more than the average of 458 on weekends. Most scam reports were submitted on Tuesdays, and the least on Sundays.

4.1.3 Most Common Scam Types

Figure 4.4 shows a breakdown of all 4,554 scam reports by scam types. The top three most common scam types being reported about were impersonation scam, online purchase scam and internet love scam. The number of scam reports classified as impersonation scams was almost double of those classified as online purchase scams. This reinforces the earlier discussion in Section 3.3 about the broad definition of impersonation scams. In particular, many scam encounters would have likely entailed elements of impersonation by perpetrators in attempts to gain victims' trust. Figure 4.4 also reveals an imbalance of classes in our text corpus, a problem which we will mitigate in Chapter 5.

We saw from the time series graphs in Figure 4.2 that the prevalence of the different types of scams fluctuated in the last four years. Figure 4.5 provides further evidence of this from a different perspective. Figure 4.5 illustrates the top five types of scams reported on 'Scam Alert'

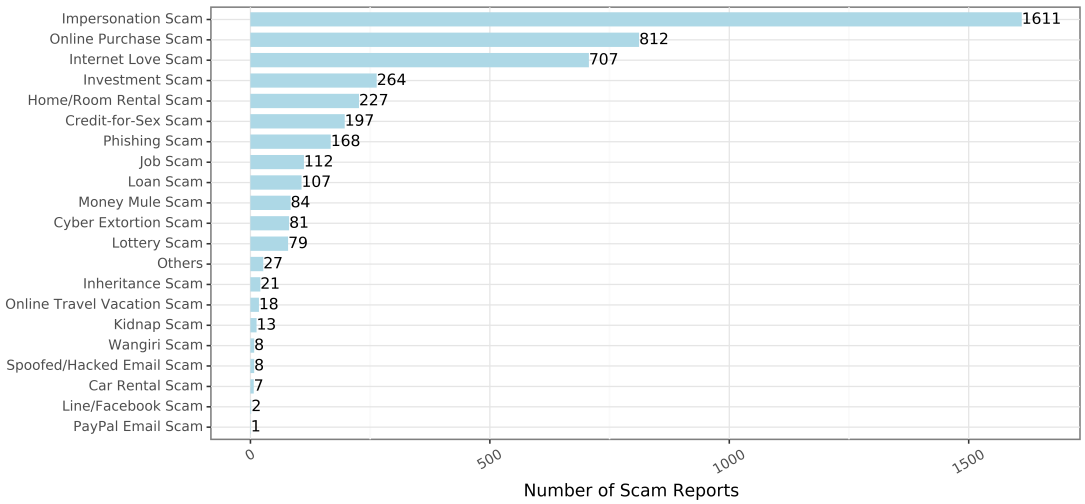


Figure 4.4: Frequency distribution of scam reports across scam types

between 2016 and 2020. There had been changes in the composition of the top five scam types every year. This points to the fact that scams were dynamic and perpetrators were always trying different ways to deceive victims. Notwithstanding, impersonation scam, internet love scam and online purchase scam were amongst the top five every year since 2015, implying that they had been most consistently prevalent.

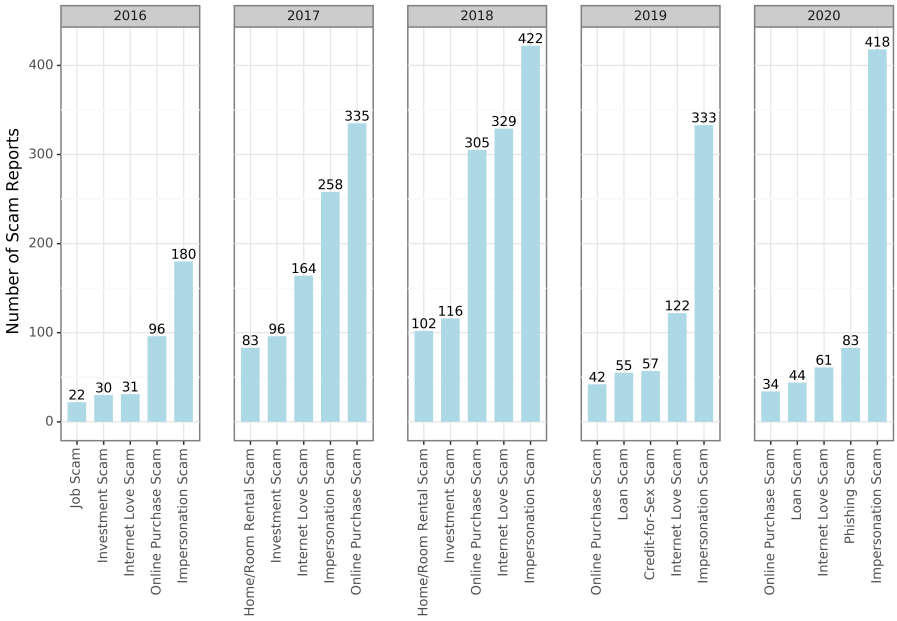


Figure 4.5: Yearly top five scam types of scam reports submitted on 'Scam Alert'

Reports relating to impersonation scams had been overwhelmingly predominant in 2020. Within the first seven months of 2020, there had been 418 reports of impersonation scams, almost equalling the total number of impersonation scam reports in the whole of 2018. In

addition, our data suggested that impersonation scam had been the dominant scam type being reported about since 2018. In contrast, official statistics from the SPF indicated that the most prevalent scam type since 2015 was online purchase scams, also referred to as *e-commerce scams* by the SPF [1–3, 5].

There could be three possible reasons behind this inconsistency. First, instead of sharing their experiences on 'Scam Alert', victims of online purchase scams might have preferred to do so directly on the online shopping platforms by writing reviews. Like published stories on 'Scam Alert', these reviews on the online shopping platforms would also be publicly accessible and possibly more effective in warning others about dubious sellers. Second, these victims were also more likely to lodge police reports in hopes of getting the SPF to trace the perpetrators, so that they might get restitution for their monetary losses. Third, there is little impetus for victims of impersonation scams to take the trouble of lodging police reports if they did not suffer monetary losses.

4.2 Exploratory Analysis of Text in Scam Reports

4.2.1 Statistical Summaries

Given that free text was at the crux of our research, exploratory analysis of the text corpus was essential. This section presents statistical summaries of the text corpus before the removal of stop words and lemmatisation.

The mean length of reports in our text corpus was 98.9 tokens, with a standard deviation of 85.4 tokens. The large standard deviation meant that there was a huge variance in the length of scam reports. This was unsurprising because reports were written by different victims. In our case, quantiles-based statistical summaries such as median and Inter-Quartile Range (IQR) were more appropriate because they were less sensitive to outliers and large variances. The median length of reports was 84 tokens and the IQR was 42 tokens.

The boxplots in Figure 4.6 summarise the distribution of length of reports by scam types. The presence of numerous outliers, shown by the black dots, reinforced the fact that there was huge diversity in the number of words used by victims to describe their scam experiences. One conjecture would be that scam types which had outliers were more multi-faceted, in that, there were many different ways in which victims were scammed. Therefore, the greater the complexities surrounding a particular scam type, the more variety there was in how victims described their stories. In contrast, scam types which did not contain outliers, such as loan scam, car rental scam and kidnap scam, were likely to be more straightforward in their modus operandi.

Reports classified as loan scams not only had the longest median length but also largest spread of report lengths in terms of IQR. On the contrary, reports classified as kidnap scam and PayPal email scam had the shortest descriptions. The longest scam report in the dataset was that of a lottery scam, containing 1,792 tokens. As Figure 4.6 shows, this was an outlier itself

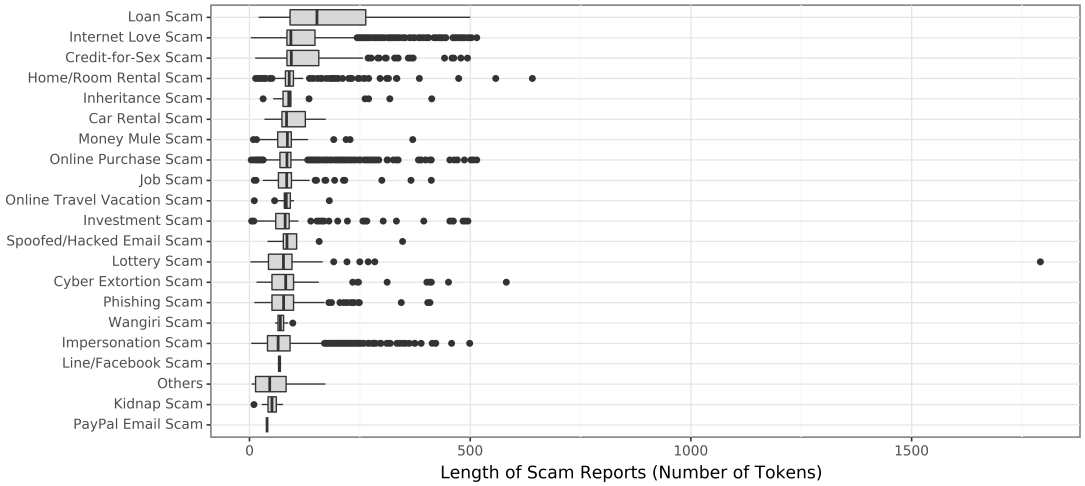


Figure 4.6: Distribution of lengths of scam reports by scam types

amongst the outliers. Notwithstanding, the majority of scam types consisted of reports with median lengths between 65 and 95 tokens.

4.2.2 Most Common and Important Tokens

This section examines the top 10 tokens in our corpus based on the following configurations:

- **Configuration 1**: Token frequency before removing stop words;
- **Configuration 2**: Token frequency after removing stop words and lemmatisation; and
- **Configuration 3**: Importance of tokens by Term Frequency-Inverse Document Frequency (TF-IDF)[2].

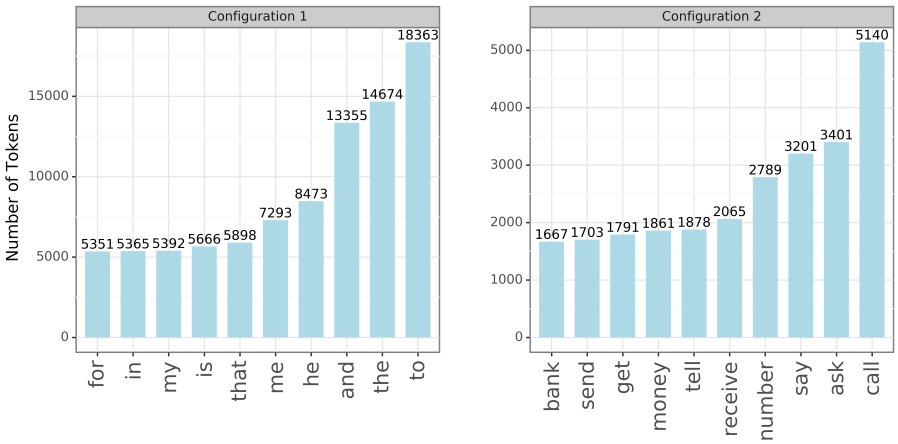


Figure 4.7: Top 10 tokens for Configurations 1 and 2

[2]The TF-IDF approach measures the importance of a particular token to a set of scam reports.

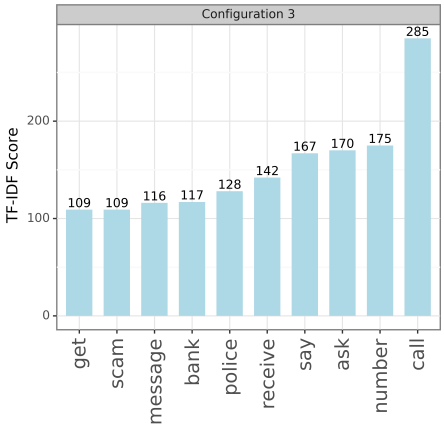


Figure 4.8: Top 10 tokens for Configuration 3

Figures 4.7 and 4.8 summarise the top 10 tokens under each configuration. In Configuration 1, there were 450,526 tokens altogether. The most commonly-occurring tokens were "to", "the" and "and", appearing 18,363, 14,674 and 13,355 times respectively. All the top tokens in Configuration 1 were stop words. With stop words removed in Configuration 2, the total number of tokens reduced almost by half to 228,704. Of these, 11,781 were unique tokens. As Figure 4.7 shows, removing stop words brought to the fore tokens which better reflected the context of our corpus. The most common tokens were "call", "ask" and "say", occurring 5,140, 3,401 and 3,201 times respectively.

High token frequency did not necessarily equate to importance. Instead of finding the most common tokens, an alternative method was to find tokens which were most unique to a set of documents. This was done using TF-IDF in Configuration 3. In general, TF-IDF places greater weights on tokens which appear infrequently across the documents. The less frequent a token appears across documents, the more unique it is. Further details about TF-IDF will be presented in Chapter 7. The top 10 key tokens by TF-IDF scores are shown in Figure 4.8. Notably, seven out of the top 10 tokens in Configuration 3 were also amongst the top 10 tokens for Configuration 2. Given the diversity of text in our dataset, there was utility in using TF-IDF to generate key tokens for individual scam types. This is shown in Table 4.1. Evidently, these key tokens provided greater insights into the defining characteristics of each scam type. These tokens can also be used to distinguish between scam types.

Table 4.1: Key tokens by scam types using TF-IDF

Scam Type	**Key Tokens by TF-IDF**
Impersonation Scam	call, number, police, receive, ask, voice, message, hang, press, phone
Phishing Scam	call, number, singtel, bank, ask, receive, internet, email, scam, claim
Home/Room Rental Scam	room, email, rent, paypal, ask, transfer, send, number, rental, work
Internet Love Scam	ask, money, send, pay, claim, meet, tell, get, bank, call
Kidnap Scam	daughter, bank, message, kidnap, report, receive, ransom, kill, call, police
Credit-for-Sex Scam	ask, call, pay, girl, meet, tell, number, money, transfer, service

Online Purchase Scam	item, email, transfer, bank, send, ask, seller, account, receive, number
Job Scam	job, pay, ask, get, number, call, work, money, company, email
Investment Scam	money, call, company, investment, account, invest, get, scam, url_link, ask
Cyber Extortion Scam	video, call, email, send, get, facebook, number, contact, receive, ask
Loan Scam	loan, transfer, ask, number, account, money, call, tell, need, message
Money Mule Scam	ask, money, call, pay, send, account, singapore, loan, number, tell
PayPal Email Scam	send, want, make, buyer, claim, close, email, fake, image, malaysia
Inheritance Scam	bank, account, money, transfer, email, contact, pay, fund, tell, receive
Lottery Scam	call, number, ask, bank, account, prize, receive, scam, tell, win
Spoofed/Hacked Email Scam	email, account, click, transfer, payment, link, money, receive, make, ask
Online Travel Vacation Scam	ticket, call, send, day, find, purchase, pay, scam, time, visa
Car Rental Scam	car, pay, company, website, book, ask, apply, course, rental, mobike
Others	call, email, website, url_link, name, scam, caller, real, pron, info
Wangiri Scam	call, number, scam, since, message, bank, back, receive, charge, handphone
Line/Facebook Scam	account, facebook, give, friend, hack, page, bill, get, ask, message

4.3 Summary

In this chapter, we derived several insights on the underlying trends in our data. These trends pertained to the usage of 'Scam Alert' as a reporting platform for scam victims, relative prevalence of different scam types according to scam report submissions as well as the submissions of scam reports by year, month, day of the week and daily average. Additionally, we explored textual information in terms of lengths of scam reports as well as the most common and important tokens by scam types. Having gained a better understanding of our text corpus, we are now in a better position to apply machine learning algorithms, which we will begin to do with supervised classification of scam reports in the next chapter.

Chapter 5

Supervised Multi-Class Classification of Scam Reports with Deep Learning

5.1 Introduction

We saw, from the literature review in Chapter 2, that there were several published studies on applications of machine learning and NLP methods using free text in the domain of cyber-crimes for classification tasks [26–33]. However, not only were these studies focused on phishing scams, they were also limited to the task of detection, which is a binary classification problem. In fact, to our knowledge, no previous studies had explored multi-class classification of scams.

Our work presented in this chapter sought to address these research gaps. More precisely, this chapter describes our work in applying various deep learning models to classify scam reports of multiple categories in free text. In doing so, we aimed to answer the research question, "Given textual description of a scam report, which category of scam types does it belong to?" The originality of this work is that it explored not only different ways of dealing with an imbalanced text data and their effectiveness in multi-class classification, but also how different types of deep learning models performed in classifying a scam report.

This chapter starts by providing a brief introduction of machine learning and the different types of deep learning models used. This is followed by an explanation of how our models were trained and evaluated for different experimental set-ups. Thereafter, we present and discuss results of our experiments, with the objective of selecting the best classification model. Finally, we highlight limitations of our work and propose alternative approaches.

5.1.1 Machine Learning and Deep Learning

Machine learning is a sub-field of AI that trains machines to learn from data without the need to be "explicitly programmed" [60]. Machine learning algorithms can be categorised into supervised and unsupervised learning. Supervised learning refers to training a model with labelled

data to predict unknown variables. Classification of scam reports, which is the focus of this chapter, is an example of a supervised learning task. On the other hand, unsupervised learning means training a model without supervision or labelled data, allowing natural discovery of patterns. One common unsupervised learning algorithm is clustering [61].

The crux of machine learning is in the representation of inputs as *features* [62]. Features used to be hard-coded in traditional machine learning, but deep learning has made it possible for features to be automatically learned [62]. Deep learning is a rapidly evolving field in machine learning. Many state-of-the-art techniques had been developed in recent years to perform various NLP tasks, including machine translation, sentiment classification and speech recognition. In subsequent paragraphs, we will give an overview of an Artificial Neural Network as well as other deep learning models used in our work.

Artificial Neural Network

An Artificial Neural Network (ANN) is the basic building block of many deep neural networks. It consists of inter-connected nodes known as *neurons*. A schematic diagram of an ANN is shown in the left image of Figure 5.1. The right image illustrates the same ANN, but in a way that is more consistent with how Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) and Bidirectional Long Short-Term Memory (Bi-LSTM) will be represented and described in subsequent paragraphs.

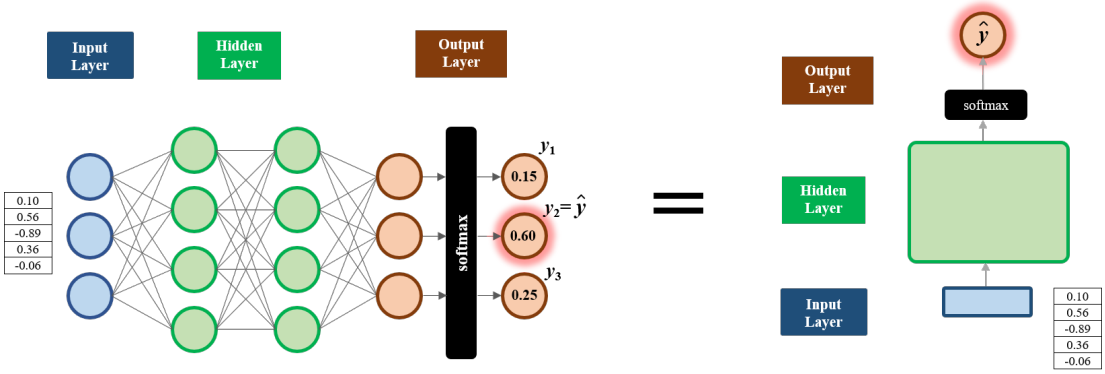


Figure 5.1: Schematic diagrams of an ANN

The neurons are arranged in layers: the input, hidden and output layers. First, input features are fed into the input layer and propagated forward through the hidden and output layers. As they propagate forward, their values change according to weights stored in the connections between the neurons. During the training process, these weights are optimised by minimising a loss function through a process known as *back-propagation*. In multi-class classification, the loss function is *categorical cross-entropy*, defined as

$$\text{Loss} = -\sum_{i=1}^{N} y_i \cdot \log \hat{y}_i, \tag{5.1}$$

where $\hat{y}_i$ is the predicted i-th value in the output layer, y_i is the target value of the corresponding label and N is the number of neurons in the output layer. In classification problems, the number of neurons in the output layer equals the number of classes. Values from the output layer are transformed into probabilities using the *softmax* function, given by,

$$\text{Softmax}(\hat{y}_i) = \frac{\exp(\hat{y}_i)}{\sum_{j=1}^{N} \exp(\hat{y}_j)}, \tag{5.2}$$

where $\hat{y}_i$ is the predicted i-th value in the output layer and N is the number of classes.

Recurrent Neural Network

The RNN is a type of ANN that is suitable for sequentially-ordered data of variable lengths, such as time series and text [62]. It consists of multiple time-steps depending on the length of input sequence. Figure 5.2 shows a simple illustration of a RNN for a classification task. The detailed architecture of a RNN cell and the underlying equations are presented in Appendix D.

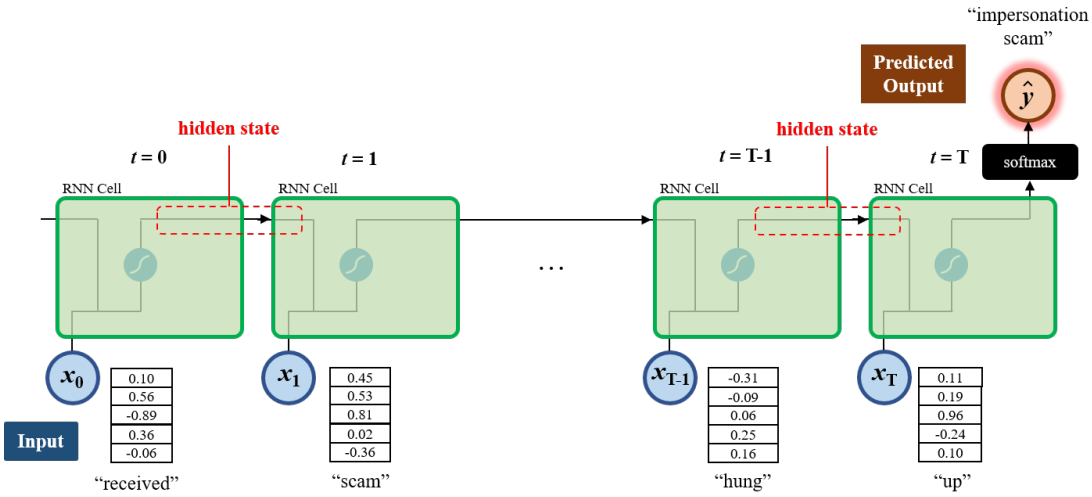


Figure 5.2: A simplified RNN for classification

Consider the following truncated input sequence: "received scam call today... she then hung up". The vectorised representation, also known as *word embedding*, of the first word, "received", is fed into the input layer at time-step $t = 0$. This is followed by the word embedding of "scam" at $t = 1$ and so on, until that of the last word, "up", at $t = \text{T}$. A hidden state runs through the entire RNN. At each time-step, the hidden state is updated using the input at each time-step and the previous hidden state. The final hidden state contains information from all previous time-steps and uses them to produce output values. These values are then converted to probabilities using softmax, and the class corresponding to the highest probability is the predicted class, denoted as $\hat{y}$.

One limitation of RNN is in capturing long-term dependencies in text [63]. In the sentence "I met a girl on Tinder... turns out to be a scammer. Beware of her!", there is a dependency

between the words “her” and “girl”. To learn the word “her”, information from much further back in the input sequence is needed. RNNs are not very good at capturing such information due to the vanishing gradients problem[1].

Long Short-Term Memory

Compared to a RNN, a LSTM is more effective in learning long-term dependencies. This is because in addition to a hidden state, a LSTM consists of a cell state running through the entire network. The cell state acts as a conveyor belt, deciding what information to carry from previous time-steps to the next. More details about the architecture and the mathematical operations involved in a LSTM cell are shown in Appendix E. In essence, a LSTM cell consists of three gates — forget gate, input gate and output gate. These gates help to regulate the flow of information using sigmoid functions defined by

$$f(x) = \frac{1}{1 + \exp(-x)}. \tag{5.3}$$

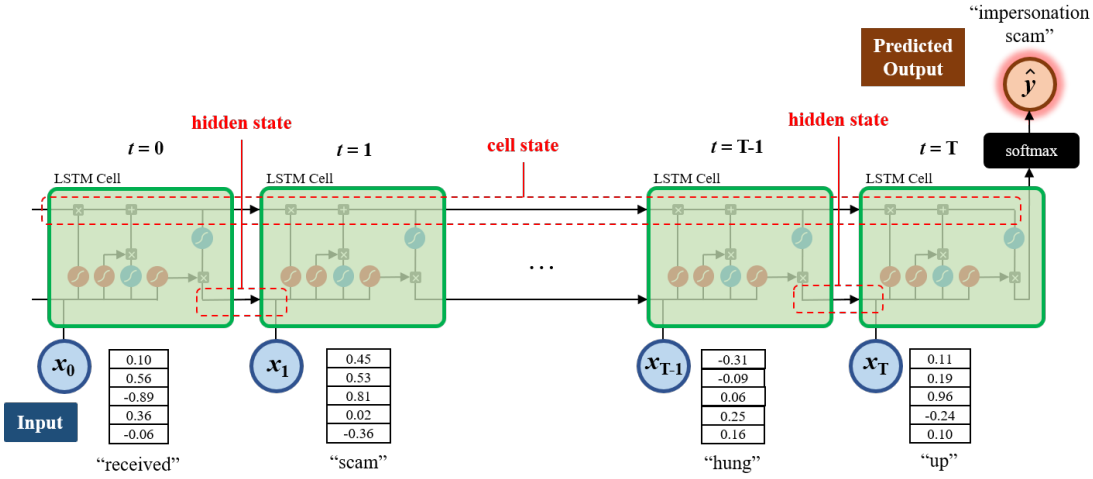


Figure 5.3: A simplified LSTM for classification

Figure 5.3 is a simple illustration of a LSTM used for classification. During training, input sequences are fed into the LSTM time-step by time-step, as with the case for RNN. The main difference is that at the final time-step, output values are generated using not only its hidden state but also the cell state. Therefore, predictions made by LSTM take into consideration information from earlier parts of the input sequence, allowing them to be more effective in learning long-term dependencies. The output values are then similarly converted to probabilities using softmax.

[1]During back-propagation, the weights in the connections between neurons are updated according to the partial derivatives of losses at each neuron. In RNNs, these gradients decrease exponentially as losses back-propagate to earlier layers of the network, making it difficult to change and optimise values of the weights.

Bidirectional Long Short-Term Memory

The RNN and LSTM are unidirectional, which means that they only utilise inputs and hidden states from earlier time-steps when making predictions. Sometimes, information from the later parts of an input sequence may be useful for the model to learn. This is the motivation behind a bi-directional structure of a RNN or LSTM.

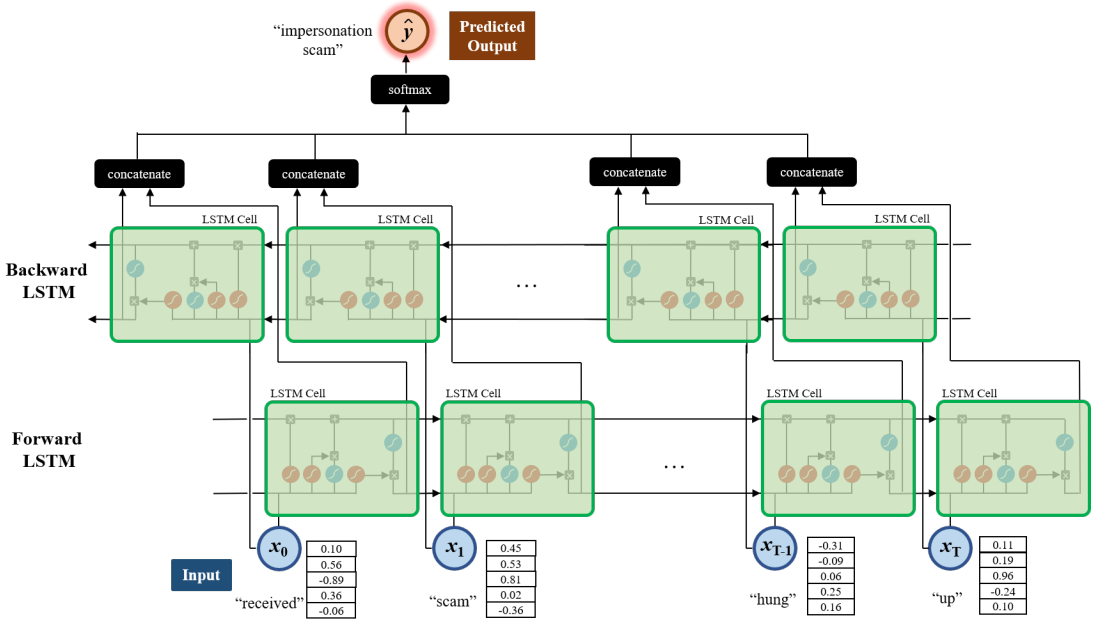


Figure 5.4: A simplified Bi-LSTM for classification

Figure 5.4 shows a simple Bi-LSTM network for classification. It consists of two LSTMs. The input sequence is fed into the forward LSTM in the normal sequence, and into the backward LSTM in the reverse sequence. The outputs of each time-step for both forward and backward LSTMs are concatenated together before output probabilities are generated using softmax.

5.1.2 Mitigating Class Imbalance

As with many real-world datasets, our dataset contained class imbalance[2]. This is evident from the frequency distribution of scam reports we saw in Chapter 4. There were 21 scam types, but the most common category, impersonation scam, constituted more than 35% of all scam reports. A skewed distribution like this poses problems to machine learning algorithms, as they tend to be biased towards the majority classes [64, 65].

In building a scam classifier, there were two possible approaches. One was to use the entire dataset and train a classifier to predict all 21 scam types; another was to train a classifier to predict only the top few scam types. There were trade-offs between these two approaches. The

[2]Class imbalance means classes or categories in the data contain an unequal number of records.

former would be trained to classify all 21 scam types, but risk having poor predictive performance on minority classes. The latter, on the contrary, restricts the number of scam types we can classify, but would be less biased towards the majority class. Given that not all scam types were equally prevalent, the latter option was adopted in our research. Specifically, we focused on the top six scam types. In addition, amongst several well-researched methods of mitigating class imbalance, we explored two: text augmentation and Synthetic Minority Over-Sampling Technique.

Text Augmentation

Text augmentation refers to creating synthetic text from authetic text. In our research, we applied three techniques adapted from Wei and Zou [66]. The first was random swap, where positions of n words in a sentence were randomly swapped. The second was random deletion, where random words in a sentence were deleted with a probability of p. The third was random insertion, which entailed inserting n random words into an original sentence. Table 5.1 contextualises these techniques with a simple example. Section 5.2.1 will further describe how these techniques were used in practice to prepare training data for our models.

Table 5.1: Examples of text augmentation

Technique	**Text**
Nil (Original Text)	"I received a call with an automated voice claiming to be from the Singapore Police Force."
Random swap	"Police claiming I call with a automated voice received to be from the Singapore an Force."
Random deletion	"received with an automated voice claiming the Singapore Police Force."
Random insertion	"I received a call law with an automated voice law claiming to be from the Singapore Police Force."

Synthetic Minority Over-Sampling Technique

The Synthetic Minority Over-Sampling Technique (SMOTE) was first proposed by Chawla *et al.* [67] in 2002 as a method of "creating synthetic minority class examples". Figure 5.5 shows a simplified illustration adapted from [68]. It shows the mapping of a set of data points on a two-dimensional vector space. First, consider the random minority sample labelled as 'A'. Next, SMOTE finds its k nearest neighbours. The default implementation of SMOTE uses $k = 5$, but we will consider $k = 2$ for simplicity. The two neighbours correspond to 'B' and 'C'. One of these neighbours is randomly selected, in this case 'B'. New samples are generated from any point along the line segment connecting 'A' and 'B'. In Section 5.2.1, we will further explain how SMOTE was used in our experiments.

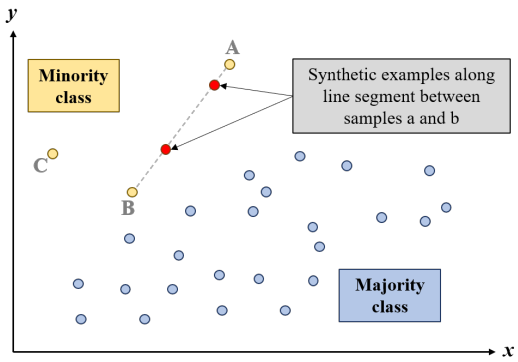


Figure 5.5: An illustration of SMOTE

5.1.3 Word Embeddings and Transfer Learning

In Section 5.1.1, we stated that words of an input sequence are fed into a RNN, LSTM or Bi-LSTM as word embeddings. These embeddings are dense vectors containing floating numbers which represent meanings of words. The length of a word embedding is equivalent to its dimension size [69] and the dimensions may be regarded as features defining that word. It is these dense word embeddings that enable algorithms to interpret and understand natural language.

The concept of word embeddings was first introduced by Bengio *et al.* [70] in 2001. More recently, there had been significant strides in pre-training word embeddings on large datasets. One example is *word2vec* embeddings, developed by Mikolov *et al.* [71] from Google in 2013. The word2vec embeddings were trained on Google News dataset containing about 100 billion words. Further details about the word2vec algorithm will be presented in Chapter 6. In 2014, the NLP team from Stanford University introduced another type of word embeddings known as *Global Vectors (GloVe)*. GloVe word embeddings were trained based on co-occurrences of words in large text corpora like Wikipedia and the English Gigaword Fifth Edition[3] [73]. The intuition is that if two words co-exist many times, it is likely that they have similar meanings [73]. It is the pre-trained GloVe embeddings that were used in our experiments.

Compared to learning word embeddings from scratch, using pre-trained word embeddings like word2vec and GloVe has numerous advantages. As the word embeddings are already trained, only weights in the final layers of the network need to be optimised, which generally translates to lesser training time. Moreover, since pre-trained word embeddings already capture meanings of words, models are known to produce better performance, particularly if the domain of application is similar to that in which the embeddings were pre-trained. Using pre-trained embeddings is also advantageous when there is insufficient training data for models to learn embeddings effectively from scratch. This idea of leveraging word embeddings trained in one domain and applying them in another domain is known as *transfer learning*. In Section 5.3, we

[3]The English Gigaword Fifth Edition is an archive of text data from English newswire that had been compiled over several years at the University of Pennsylvania [72].

will discuss the impact of transfer learning on our experiments.

5.2 Methodology

This section describes the methodologies involved in training RNN, LSTM and Bi-LSTM for multi-class classification of scam reports. We used `keras` [74] and `sklearn` [75], popular Python libraries for deep learning and machine learning. Cognisant of the class imbalance problem in our data, we experimented with the following three different set-ups:

1. **Experiment 1**: Classification with imbalanced classes;
2. **Experiment 2**: Classification with balanced classes using text augmentation; and
3. **Experiment 3**: Classification with balanced classes using SMOTE.

5.2.1 Training the Models

Experiment 1: Classification with Imbalanced Classes

In the first part of this sub-section, we describe our methodology in preparing the training data for Experiment 1. As stated in Section 5.1.2, our approach was to classify the top six scam types. We first extracted scam reports belonging to these categories. There were 3,818 scam reports, the breakdown of which is shown in Table 5.2. These scam reports, or training examples, contain fully pre-processed text[4] as well as the corresponding scam types or labels. These 3,818 records were partitioned — 80% or 3,054 examples as the training set and and the remaining 20% or 764 examples as the test set. Figure 5.6 illustrates our steps in preparing the training data.

Table 5.2: Breakdown of top six scam types

Scam Type	Count
Impersonation Scam	1611
Online Purchase Scam	812
Internet Love Scam	707
Investment Scam	264
Home/Room Rental Scam	227
Credit-for-Sex Scam	197

Given that the labels were originally in the form of text, each label was assigned a unique number between 0 and 5. Next, the fully pre-processed text was tokenised and each token converted to a number corresponding to its index. To train our deep learning models, the input sequences needed to be of fixed lengths. Thus, the next step was to pad or truncate sequences according to a specified length. Input sequence length was one parameter we needed to set. Too long a sequence length and many input sequences would mostly contain '0's. Short sequence

[4] In this dissertation, the text is considered "fully pre-processed" if it had undergone the entire text pre-processing pipeline described in Section 3.5, including removal of stop words and lemmatisation.

Input		Labels
"I met a girl on Tinder. Turned out to be a scam"		Internet Love Scam
"Received a scam call today from a lady claiming to be from Microsoft. She said my Internet had suspicious activities."		Impersonation Scam
"I paid for hand sanitisers but never received it. Stupid scam!"		Online Purchase Scam

Convert text input and labels to numbers

Input		Labels
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 3, 11]	**Length = 12 tokens**	3
[12, 3, 11, 13, 14, 15, 3, 16, 17, 9, 10, 15, 18, 19, 20, 21, 22, 23, 24, 25]	**Length = 20 tokens**	2
[1, 26, 27, 28, 29, 30, 31, 12, 32, 33, 11]	**Length = 11 tokens**	5

Pad or truncate input sequences

Input		Labels
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 3, 11, 0, 0, 0, 0]	**Length = 16 tokens**	3
[12, 3, 11, 13, 14, 15, 3, 16, 17, 9, 10, 15, 18, 19, 20, 21]	**Length = 16 tokens**	2
[1, 26, 27, 28, 29, 30, 31, 12, 32, 33, 11, 0, 0, 0, 0, 0]	**Length = 16 tokens**	5

Figure 5.6: Steps involved in preparing data for model training

lengths, on the other hand, would entail information loss as longer sequences would need to be truncated. To strike a balance, our approach was to set the input sequence length as the 80th percentile of the distribution of token lengths in the entire text corpus. In Experiment 1, this equalled 55 tokens.

For each of the three experiments, we trained the following six different models. All experiments were performed on Google Colaboratory, a cloud service that provides free Graphics Processing Unit (GPU)[5].

- RNN, without GloVe word embeddings;
- RNN, with GloVe word embeddings;
- LSTM, without GloVe word embeddings;
- LSTM, with GloVe word embeddings;
- Bi-LSTM, without GloVe word embeddings; and
- Bi-LSTM, with GloVe word embeddings.

Before training our models, we first defined their architectures. The detailed architectures are presented in Appendix F, but we will elaborate on the RNN architecture using Figure 5.7. The first layer was the *input layer*, which took in input sequences of a fixed length. The next layer was the *embedding layer*, which was initialised with random weights at the start of training. As the model trained, these weights were updated. Where pre-trained GloVe embeddings were used, only embeddings of words not present in the GloVe database were learned from scratch.

[5] A GPU is a specialised processor that has an optimised memory bandwidth to handle large amounts of computations. It is much more efficient than standard Central Processing Unit (CPU) in training deep learning models.

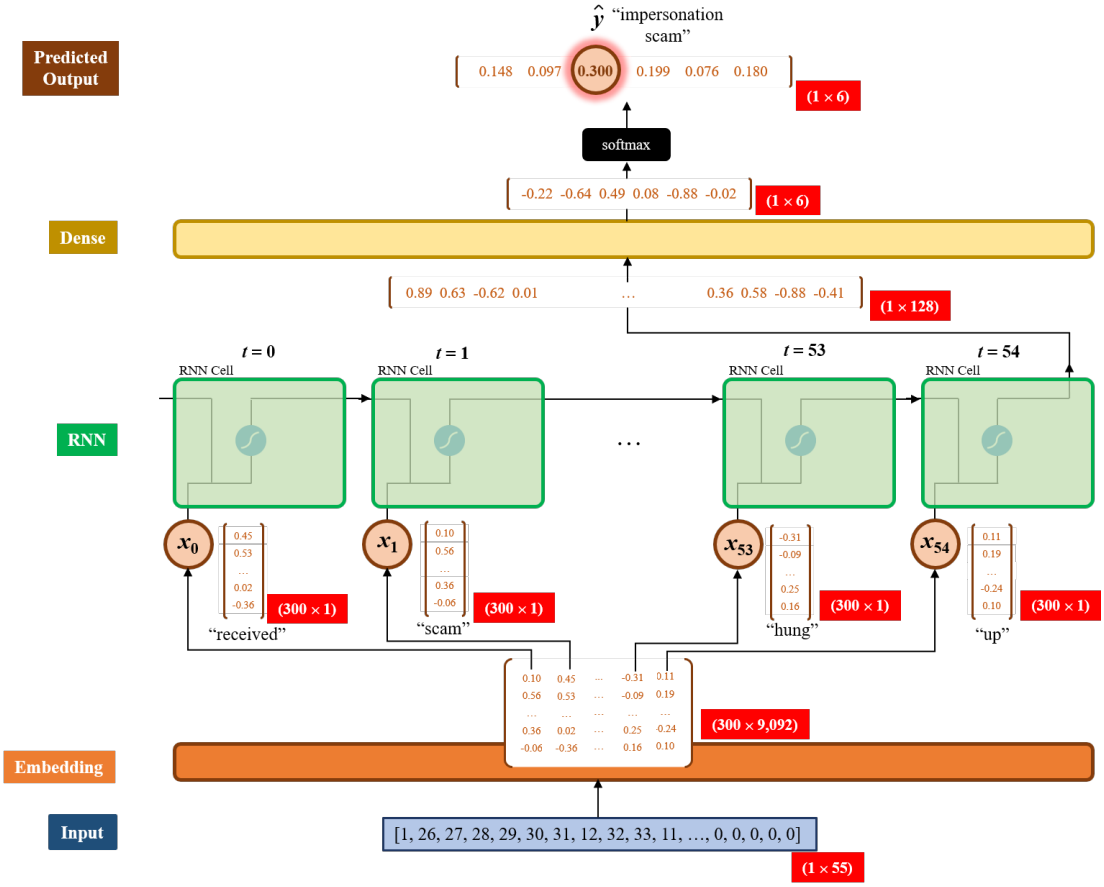


Figure 5.7: Architecture of RNN used in model training

The output from the embedding layer was an *embedding matrix*, whose shape was dimension size by the size of vocabulary. With a vocabulary size of 9,092 and our dimension size chosen as 300, the shape of the embedding matrix was 300 × 9,092 and each word embedding was a 300 × 1 vector.

After the embedding layer, each word of an input sequence was sequentially fed into the RNN at each time-step as a 300 × 1 embedding. The output from the RNN was a one-dimensional vector containing a specified number of internal units. In our models, the number of internal units was set as 128. The shape of the output from the RNN was therefore 1 × 128. In the case of the Bi-LSTM, the shape of the output is 1 × 256, since two 1 × 128 were concatenated at each time-step. The final layer was the *dense layer*, which fully connected the 128 internal units to six output units. The six output values from the dense layer were converted to probabilities using softmax and the class corresponding to the highest probability was the predicted class.

In training each of the six models, we implemented five-fold cross-validation as a way to mitigate over-fitting[6]. This is illustrated in Figure 5.8. As previously stated, our text corpus was

[6]Over-fitting occurs when a model is fitted perfectly to the training data such that it has learned random noises in the data rather than general patterns. Over-fitting affects the ability of a model to generalise on unseen data [61].

divided into training and test sets by a 80:20 ratio. In each iteration, the training set containing 3,054 training examples was partitioned into five folds, four of which were used to train the model ("training") and the remaining fold to validate the trained model ("validation"). After every iteration, a different fold was used for validation. During training, the categorical cross-entropy loss function was used with the Adam optimiser[7] and a learning rate of 0.001 to optimise the weights in the layers. At the end of the 5th iteration, there were five different models with slightly different predictive performances on the test set. These were then used to derive the average performance of the model.

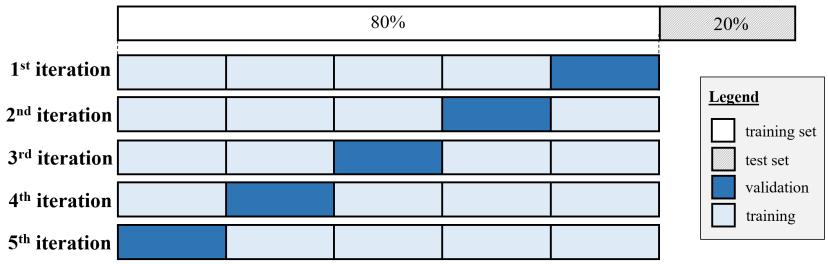


Figure 5.8: Splitting of data for five-fold cross validation

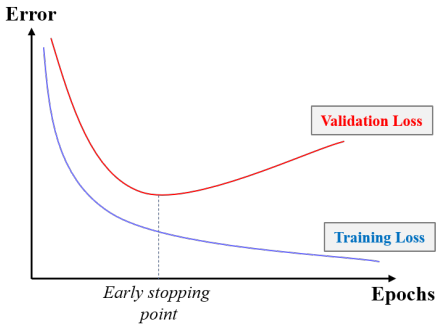


Figure 5.9: An illustration of early stopping

A consistent approach was taken to decide when the training of each model should stop. This involved comparing the *training loss*, which is the loss with respect to the training set, against *validation loss*, which is the loss with respect to the validation set. This is illustrated in Figure 5.9. These losses were computed using the categorical cross-entropy loss function. In general, training loss will always decrease with the number of epochs, but validation loss will decrease initially before increasing at some point. The increase in validation loss is a sign that the model is starting to over-fit [77]. In our research, we adopted the guidance of Bishop [77] to stop the training at the point of minimum validation loss. This is also known as *early stopping*.

Another step taken to reduce over-fitting was *dropout*, a technique developed by Srivastava *et al.* [78] in 2014. It refers to the random disappearance of neurons from the neural network

[7]The Adam optimiser is a stochastic gradient descent optimisation method that is suited for problems with a large number of parameters, easy to implement and computationally efficient [76].

during training. In his paper, Srivastava *et al.* showed that dropout prevented neurons from "co-adapting too much", which helped to reduce over-fitting [78]. Two types of dropouts were implemented in our models: normal dropout between layers of RNN or LSTM; and recurrent dropout between each RNN or LSTM cell.

The Stanford NLP team had made publicly available GloVe word embeddings trained on different text corpora and of different dimension sizes. The GloVe embeddings used in our models were of 300 dimensions, pre-trained on the Wikipedia and the English Gigaword Fifth Edition text corpora. Using GloVe embeddings meant that there were lesser parameters to be trained. For instance, in the RNN model with GloVe embeddings, only 55,686 parameters needed to be trained, as compared to 2.78 million parameters in the model without GloVe embeddings. A summary of parameters used in all three experiments is presented in Appendix G.

Experiment 2: Classification with balanced classes using text augmentation

Experiment 2 involved a different methodology in preparing training data. We first set a threshold for the number of training examples per scam type. In our case, this threshold was set as 400. For scam types with at least 400 training examples, we randomly sampled 400 training examples. For scam types with the number of training examples less than 400, we performed text augmentation to make up for the deficit. The result was a balanced dataset in which every scam type contained 400 training examples.

As described previously, the three text augmentation methods used were random swap, random deletion and random insertion. The scam types for which text augmentation was performed were investment scam, home/room rental scam and credit-for-sex scam. From each of these scam types, we first randomly selected one authentic scam report. For each sentence in the randomly selected scam report, one of the three text augmentation techniques was randomly chosen and applied. In the case of random swap, positions of four random words were swapped. For random deletion, random words were deleted with a probability of 0.30. As for random insertion, two random words were inserted at random positions.

Our main consideration in using a variety of text augmentation techniques with these parameters on a single text was to strike a balance between generating synthetic text that is too similar to the authentic reports and that which is too diverse such that they carry vastly different meanings. Nonetheless, it would be a reasonable extension of this work to investigate how quality of augmented text affects model performance.

This process of augmenting text from authentic scam reports was repeated until the number of records in each scam type reached the set threshold of 400. Thereafter, stop words were removed before tokens in the text were lemmatised. It was the lemmatised version of the text that formed our training data for Experiment 2.

There were two key differences between Experiments 1 and 2 with respect to model training. One, the vocabulary size in Experiment 1 was 9,092, compared to 6,992 in Experiment 2. The

second difference was the input sequence length, which we had set as 80% percentile of the lengths of all scam reports in the training corpus. This was 55 tokens for Experiment 1, but 66 tokens for Experiment 2.

Experiment 3: Classification with balanced classes using SMOTE

Similar to Experiment 1, Experiment 3 also utilised scam reports extracted from the top six scam types shown in Table 5.2 as the main training data. Using the SMOTE implementation in the Python library `imblearn` [79], we performed over-sampling of the three scam types whose number of training examples was below the threshold of 400, and a random under-sampling of the other three. Since SMOTE operates in the "feature space instead of the data space" [67], the training data it used was no different from that of Experiment 1, which meant that the vocabulary size and input sequence length were also the same — 9,092 tokens and 55 tokens respectively. Additionally, the methodology in model training was also the same as in Experiment 1.

5.2.2 Evaluating the Models

This sub-section describes the approach in evaluating our models in all three experiments. We briefly stated in the previous section that five-fold cross-validation produced five models with different performance. Specifically, they had different levels of accuracy on the validation sets, or *validation accuracy*, in each iteration of the cross-validation. Amongst the five models, the one gave the highest validation accuracy was selected as the "best version". This best version was subsequently tested on the unseen test set which contained 764 test examples. A model's predictive accuracy on the test set, or commonly referred to as *test accuracy*, reflects its ability to make predictions on unseen data. To ensure a fair comparison of models across all three experiments, the same test set was used.

5.3 Results and Discussions

The following section is organised in two parts. First, we evaluate the effectiveness of GloVe word embeddings and the impact of transfer learning on our models. Next, we present and discuss results of our experiments in order to select the best model for classification of scams.

5.3.1 Assessing Effectiveness of GloVe Embeddings

There were mixed findings on the effectiveness of GloVe word embeddings. Figure 5.10 presents the results of each model, with and without GloVe embeddings, in terms of test accuracy. In Experiment 1, GloVe embeddings had a positive effect on test accuracy for all three models. However, the contrary was true for Experiment 2, where models without GloVe embeddings were more accurate. In Experiment 3, GloVe embeddings made LSTM and Bi-LSTM more

accurate than without, but this was not true for RNN. These observations pointed to a potential limitation of transfer learning. Transfer learning is about leveraging the knowledge gained by a model in one domain and applying it in a different domain. Transfer learning may not be as effective if the domains were vastly different.

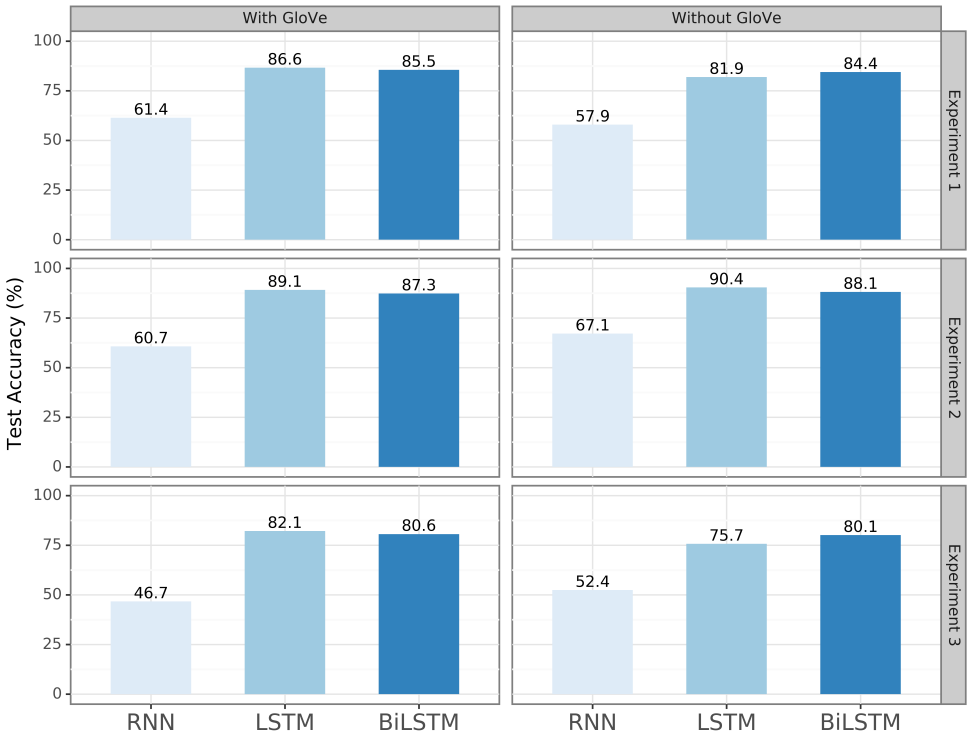


Figure 5.10: Results of experiments in terms of test accuracy

It is evident from the results that GloVe embeddings had limited effectiveness in our models' performance. This could boil down to differences in the domains of application. The GloVe word embeddings were pre-trained on Wikipedia articles and news articles from English Gigaword Fifth Edition. These corpora likely contained text which were not as messy, noisy and linguistically varied as our text corpus. Moreover, these corpora represented a wide range of genres whereas our text corpus was centered around a very niche domain — scams. It is plausible that the GloVe embeddings might not have been suited for our text corpus. The fact that some models without GloVe embeddings attained good performance suggested that they were able to learn embeddings of words specific to our text corpus effectively from scratch.

5.3.2 Selecting the Best Model

Accuracy on Test Set

From Figure 5.10, it is evident that LSTM and Bi-LSTM models produced more superior results than RNN. The average test accuracy of LSTM and Bi-LSTM models was both 84.3%, compared to 57.7% for RNN models. In addition, there were mixed results as to whether

LSTM or Bi-LSTM performed better. In general, the difference between LSTM and Bi-LSTM were mostly observed to be small. Where GloVe embeddings were used, LSTM models were marginally more accurate on the test set than Bi-LSTM models. However, the difference was less than 2%. Conversely, where GloVe embeddings were not used, Bi-LSTM was more accurate than LSTM in two out of the three experiments.

The central idea of a bidirectional structure in a Bi-LSTM was to enable the model to learn both from the past and the future of text sequences. While this bidirectional concept has shown promise in other NLP tasks like speech recognition [80], results of our experiments seemed to suggest that Bi-LSTM was on par with a simple LSTM, at least for our domain of multi-class text classification. We speculate that input sequence length may have a role to play in the performance of LSTM and Bi-LSTM models. On one hand, a longer input sequence means a bigger window of text for a Bi-LSTM to be trained on. Assuming that the latter parts of the text contain useful information about scams, a Bi-LSTM will be a more sensible choice. On the other hand, given our noisy and unstructured text of hugely varied lengths, a bigger window may not necessarily yield more meaningful information. Further experimentation is warranted to study how input sequence length will affect the performance of both LSTM and Bi-LSTM in multi-class classification. Only then will we find a "sweet spot" in terms of an input sequence length that generates the best test accuracy.

Precision, Recall and F1-Score on Test Set

The drawback of using test accuracy as the sole evaluation metric is that it only examined the proportion of all correct predictions made by a model, without considering the number of observations in each class in the test set. Since test accuracy did not distinguish correct predictions by classes, how well a model predicted each class was unknown. In light of this, we used *precision* and *recall* as alternative metrics to measure a model's predictive ability.

To gain some intuition about precision and recall, consider internet love scam as an example. Precision is about asking, "Out of all reports predicted by a model to be internet love scams, how many were actually internet love scams?" On the other hand, recall asks the question, "Out of all internet love scam reports, how many did a model predict correctly to be internet love scams?" Both precision and recall range between 0 and 1; the closer they are to 1, the better. With both precision and recall values, it is difficult to compare models quantitatively. The *F1-score* mitigates this by combining precision and recall into a single number. F1-score also ranges between 0 and 1; and the closer it is to 1, the better the model is in predicting a particular class. F1-score is defined by the following expression:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{5.4}$$

Figure 5.11 shows the results of the models on the test set in terms of F1-scores. Labels indicating the F1-scores had been omitted in this figure for greater clarity. The results in terms

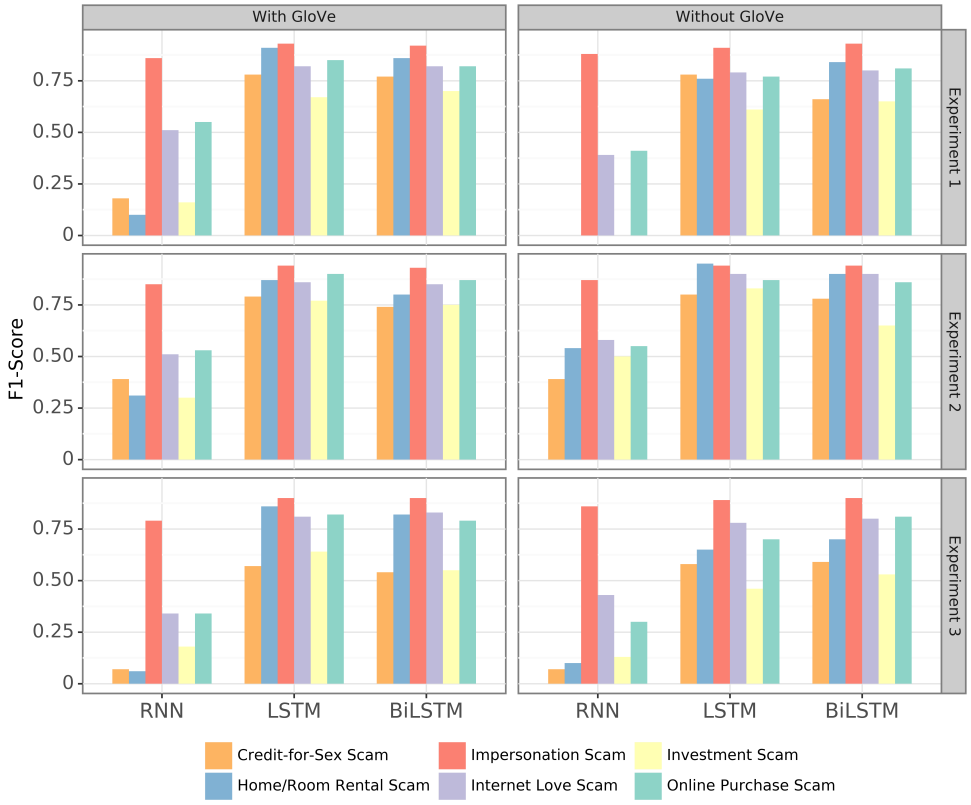


Figure 5.11: Results of experiments in terms of F1-scores

of precision and recall are reported in Appendix H. The advantage of evaluating our results using F1-scores as compared to test accuracy is that it brought to the fore valuable insights on the relative strength of the models in predicting each class. Consider, as an example, the RNN model without GloVe embeddings in Experiment 1. From Figure 5.10, we observe that the test accuracy for this model was 57.9%. For the same model, Figure 5.11 tells us that it was only able to predict three out of the six classes, which was not ideal. In Experiments 2 and 3 where classes were balanced, the models had better predictive performance on some classes than on others. Consider the Bi-LSTM model without GloVe embeddings in Experiment 2. While it attained a high test accuracy of 88.1%, Figure 5.11 shows that it was relatively inferior in predicting investment scams. Similarly, in Experiment 3, LSTM and Bi-LSTM models had overall test accuracies of at least 75% but Figure 5.11 shows that they were much weaker in predicting credit-for-sex and investment scams.

Our yardstick for selecting the best model was a strong performance on all classes. Given the results in Figure 5.11, there were two contenders: the LSTM models of Experiment 2 with GloVe embeddings and without. Both models had F1-scores of at least 0.77 on each class. In addition to F1-scores, we also examined recall scores to aid us in choosing the best model. This was because recall fundamentally reflects how well a model correctly predicts a scam report. Based on recall scores presented in Appendix I, the LSTM model of Experiment 2 without

GloVe embeddings was marginally better than the LSTM model with GloVe embeddings. Its macro-average recall[8] and weighted-average recall[9] were 0.895 and 0.904 respectively, compared to 0.872 and 0.891 for the latter. Given these analyses, the LSTM model of Experiment 2 without GloVe embeddings was assessed to be the best model.

These findings lend credit to the reliability of text augmentation as a method to counter imbalance in text data. In Section 5.2.1, we highlighted our cautious approach in creating synthetic text such that they were neither too similar to nor too different from the original text. The experimental results showed that this approach worked well. Notwithstanding, it would be premature to consider it the best approach. There are various ways to extend this work to investigate how different ways of augmenting text will affect a model's predictive performance. We can vary the number of words in a sentence whose positions are to be swapped, or the number of random words to be inserted or deleted.

Another way to extend this research is to vary the threshold. In our work, the threshold was set as 400. Based on this threshold, the number of synthetic text generated was 512[10]. An important consideration in choosing a threshold is to avoid creating too many synthetic text. Increasing the threshold means sampling more from majority classes, which will capture more training examples for the model to learn from, but more synthetic text needs to be generated for the other classes. We hypothesise that too many synthetic text may dilute the essence of scam reports and affect model performance. More research is necessary to validate this hypothesis. Our choice of 400 as the threshold was, to some extent, arbitrary. Thus, another possible area of future research is to establish a systematic method of determining a threshold that balances between sampling sufficient training examples from majority classes and avoiding too many synthetic text for minority classes.

Results of Experiment 3, as shown in Figure 5.11, revealed some limitations about SMOTE. In terms of F1-scores, the RNN models had significantly better predictive performance on impersonation scam than any other scam types, likely due to strong inherent bias towards the majority class. The disparity was much lesser in LSTM and Bi-LSTM models. It was also observed that LSTM and Bi-LSTM models were better at predicting some classes than others. They were generally better at predicting impersonation scam, online purchase scam and internet love scam, which were the majority classes, but their performance on minority classes was less optimistic. In fact, the two lowest F1-scores for LSTM and Bi-LSTM models corresponded to two of the minority classes — credit-for-sex scam and investment scam. These findings seemed to point towards an inadequacy of SMOTE as a technique to minimise bias effectively. This could be attributed to our training data being of high dimensions which, as alluded to in some literature, could fail to improve a model's performance when used with SMOTE [81, 82].

[8] Macro-average recall is the average of recall scores across all classes.

[9] Weighted-average recall is the average recall score weighted by relative number of observations in each class.

[10] From the breakdown of the top six scam types shown in Table 5.2, a threshold of 400 meant creating 136 synthetic text for investment scam, 173 for home/room rental scam and 203 for credit-for-sex scam.

5.4 Limitations

Having elucidated details of our methodology and discussed the results, we now take a broader outlook and highlight limitations in our research. There were two main limitations. The first relates to human error in manual classifications of scam reports by NCPC. Part of the vetting process after a scam report is submitted by a victim is to manually inspect the classification based on the textual description. As we discovered, 292 scam reports out of 4,554 were either misclassified or unrelated to scams. Notwithstanding the descriptions of various scam types on 'Scam Alert', human error in manual classification is not inconceivable. Given the dynamic nature of scams, the criteria and human judgement used in manual classification would likely evolve with time, giving rise to some chronological bias. Moreover, accurately classifying scam reports requires experience and knowledge, which means different annotators may classify reports differently.

Although human error is inevitable, it compromises the reliability and predictive performance of our models. It consequently inhibits the ability of the models to accurately learn features of scam reports that define each scam type and make sound predictions. The reclassification of the 292 reports during data cleaning was an effort to minimise the impact of human error, but more can be done. The actual number of misclassified scam reports could well be more than 292, since we did not inspect every single report during data cleaning. Hence, an improvement from our current work will be to dedicate more resources to manually inspect all scam reports in our text corpus to ensure that they are correctly classified. Though laborious and time-consuming, this one-off process will ultimately help our models produce more robust predictions.

The second limitation is that our chosen model had only been trained to classify the top six scam types. This was because of our decision to prioritise building an unbiased classifier over a classifier that classifies all classes but with bias towards the majority classes. As a result, any scam report that is not of the top six scam types will be classified by our model into one of the six categories anyway. This impedes the model's ability to predict classifications for lesser known scams, which is not ideal especially if the model was to be deployed into production. There are areas of improvement. Since we have gained some assurance on the efficacy of data augmentation techniques in countering imbalance, we can consider expanding the number of categories of scam types to be classified. The threshold in our experiments was set as 400, but it is also worth exploring whether varying it, whilst expanding the number of scam types to be classified, will improve model performance.

5.5 Alternative Approaches

Besides the suggestions mooted in the previous section to overcome limitations, there are several alternative ways in which our research goal of classifying scam reports can be met. In this

section, we briefly highlight four.

The first is to modify our models. The architectures of our deep learning models were considered relatively rudimentary compared to many other sophisticated applications. One possible improvement is to stack multiple layers of our RNN, LSTM or Bi-LSTM to build deeper models. Another is to develop deep learning models with attention mechanism which, in recent years, had been a game-changer in the field of NLP [83]. With attention mechanism, words which are integral to the meaning of a document contribute more to their embeddings [84].

Another alternative approach is to use other pre-trained word embeddings besides GloVe. Examples include the Bidirectional Encoder Representations from Transformers (BERT) embeddings as well as word2vec embeddings. In fact, our work presented in Chapter 6 derived word2vec embeddings for words in our text corpus. Since they already contain useful knowledge, it may be of value to use them at the embedding layers of our models in place of GloVe.

The third alternative is to use algorithms other than deep learning. Examples include Random Forest, Support Vector Machine, and Naive Bayes. Whilst there were previous studies that applied these algorithms for the binary classification task of detecting phishing scams, there were none applying them for a multi-class classification task. A key advantage of these algorithms over deep learning methods is they are computationally more efficient.

A fourth alternative approach is to dig deeper into the wrong predictions made by the models on the test set. For example, it was observed that our chosen LSTM model most commonly mistook internet love scams as online purchase scams, and online purchase scams as impersonation scams. With such information, we can do further analysis, such as the extraction of softmax probability values to uncover any patterns or hierarchy in those wrong predictions and better understand how the models' predictions can be improved.

5.6 Summary

In the work presented in this chapter, we applied three deep learning models, namely the RNN, LSTM and Bi-LSTM, in classifying scam reports in free text into multiple categories. We investigated two techniques of countering imbalance in our data — text augmentation and SMOTE. We experimented with both balanced and imbalanced datasets and also examined the efficacy of pre-trained GloVe word embeddings. Results of our experiments led us to conclude that the best-performing model was the LSTM model trained without GloVe embeddings on the dataset balanced using text augmentation.

As the first study to explore multi-class classification of scams in free text, it is hoped that our work will not only advance the understanding of the different data augmentation techniques for text data, but also lay the groundwork for future research using more sophisticated machine learning techniques for multi-class classification. Using our chosen classification model, we will show, in Chapter 7, how we can predict the scam type given a new scam report.

Chapter 6

Unsupervised Encoding of Scam Reports as Document Embeddings

Apart from supervised applications, our research also involved applying unsupervised NLP techniques on free text. Based on existing literature around the applications of machine learning in the domain of scams, only Feng *et al.* [32] and Wu *et al.* [33] had used unsupervised methods. Even then, both were for the detection of phishing scams specifically and neither had used free text in the form of natural language. Feng *et al.* [32] used document object model of websites to represent websites as vectors whereas Wu *et al.* [33] used crypto-currency transaction records. Thus, there appeared to be an insufficient understanding of how unsupervised NLP techniques can be applied on free text in natural language across different types of scams.

This inadequacy strengthened the impetus of our work presented in this chapter. As this chapter will explain, this part of our research examined how scam reports could be encoded as vectors of numbers, a concept known as *vector semantics*. These vectorised representations of scam reports were then used to answer our two remaining research questions: "Given textual description of a scam report, which other scam reports share similar modus operandi?" and "Given textual descriptions of a set of similar scam reports, what are the common characteristics of their modus operandi?"

We begin this chapter by providing a brief explanation of the word2vec and doc2vec algorithms, as well as a theoretical background of cosine similarity. Following this, we present our methodology in training doc2vec models to encode scam reports as document embeddings. We also describe how the models were evaluated. In doing so, we introduce a novel evaluation framework known as *Similarity-Dissimilarity Quotient*. We will discuss the results of our models in order to select the most optimal doc2vec model, before closing this chapter by highlighting limitations of our work and alternative approaches.

6.1 Introduction

6.1.1 Vector Semantics and Word2Vec

The concept of vector semantics traces back to the 1950s, when philosopher Ludwig Wittgenstein proposed that "the meaning of a word is its use in the language" [85, 86]. Building on this intuition, several linguists during that time postulated that a word could be defined by its context, including its "neighbouring words and grammatical environments" [86]. Vector semantics combines this with the idea of Osgood *et al.* [87] to define a word as a list of numbers [86].

One application of vector semantics is the *word-to-vector* (word2vec) model, developed by Mikolov *et al.* in 2013 [71]. There are two types of word2vec algorithms: *Continuous Bag-of-Words* (CBOW) and *skip-gram*. In CBOW, a model is trained to predict a target word given its context words. Consider the text, "I bought headphones on Carousell". As shown in the left diagram of Figure 6.1, CBOW trains a model to predict the target word, "headphones", given context words within a certain window size of the target word. Skip-gram, on the other hand, trains a model to predict context words given the target word, "headphones". Both models use a simple ANN.

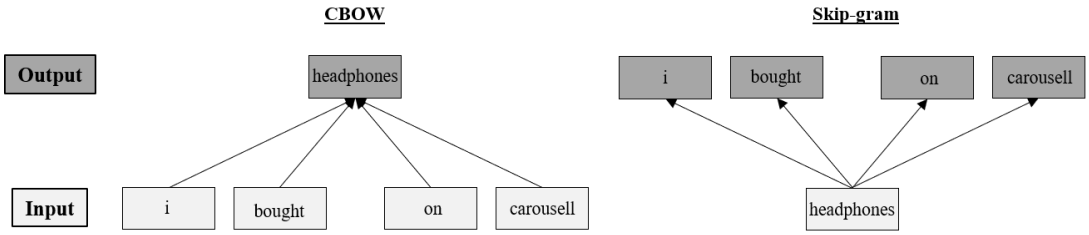


Figure 6.1: Schematic representations of CBOW and skip-gram models of word2vec

6.1.2 Doc2Vec

Vector semantics can also be applied on documents of text. The simplest way is the Bag-of-Words (BOW) approach, which represents documents by the frequency of words that occur within each document. Despite its simplicity, BOW has several drawbacks. It ignores word order, which means different sentences with the same words would have the same representations. BOW produces sparse vectors that contain mostly '0's and are computationally inefficient to process. They are also limited in their abilities to represent meanings of documents [88].

Paragraph vector developed by Le and Mikolov [88] in 2014 addresses these limitations of BOW. Paragraph vector is an unsupervised algorithm that "learns fixed-length feature representations" of text of different lengths [88]. Such text can include phrases, sentences or documents. The application of paragraph vector on documents is more commonly known as *document-to-vector* (doc2vec). Doc2vec can be thought of as an extension of word2vec. The key difference is that in addition to using only words to make predictions as in word2vec, doc2vec uses a unique

document tag identity (ID) for each document.

There are two types of doc2vec algorithms. The first is called *Distributed Memory Model of Paragraph Vector* (PV-DM). PV-DM is similar to the CBOW approach in word2vec. In PV-DM, a vector corresponding to a document ID is concatenated with vectors representing the context words from the document before being trained using an ANN to predict a target word. In Figure 6.2, the vector corresponding to tag ID #22 is trained with the vectors of the context words to predict the target word "headphones".

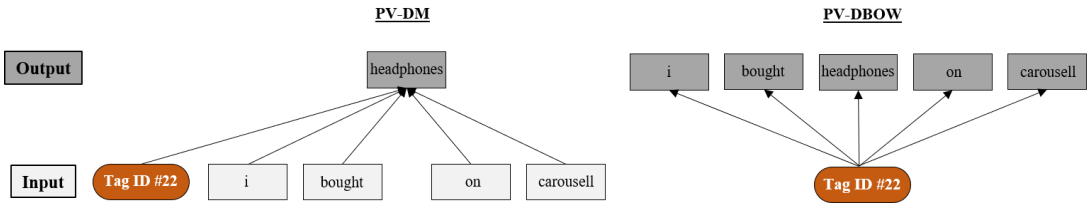


Figure 6.2: Schematic representations of PV-DM and PV-DBOW models of doc2vec

The other doc2vec algorithm is known as *Distributed Bag-of-Words version of Paragraph Vector* (PV-DBOW), which is analogous to skip-gram of word2vec. In PV-DBOW, a document vector is trained to predict words in a window of text within the document. In the example of Figure 6.2, the vector for tag ID #22 is trained to predict the words "i", "bought", "headphones", "on" and "carousell". In our research, we regarded the textual description of each scam report as a document and assigned each with a unique document tag ID. In the remainder of this dissertation, the term *document embedding* will be used to refer to the vector of numbers that represents a scam report.

6.1.3 Cosine Similarity

By representing scam reports as document embeddings, we can discover interesting semantic properties about the reports. Using *cosine similarity*, we can find scam reports which are similar to one other. Cosine similarity measures the cosine of the angle between two vectors in a multi-dimensional vector space. Mathematically, it is defined by,

$$\cos\theta = \frac{\mathbf{X} \cdot \mathbf{Y}}{\|\mathbf{X}\|\|\mathbf{Y}\|}, \tag{6.1}$$

where $\mathbf{X}$ and $\mathbf{Y}$ are vectors and θ is the angle between them. If vectors $\mathbf{X}$ and $\mathbf{Y}$ are perfectly similar, θ is 0° and cosine similarity is 1. Conversely, if vectors $\mathbf{X}$ and $\mathbf{Y}$ are perfectly dissimilar, θ is 180° and thus cosine similarity is -1. The closer the cosine similarity between two vectors is to 1, the more similar they are. To further illustrate how cosine similarity is applied, consider the following set of three scam reports:

- **Document A**: "*I bought hand santisers on ninelif.sg but took a while for me to realise that it is a scam.*"

- **Document B**: "*I purchased kids' surgical masks on ninelif.sg with payment made via PayPal. I followed up with the seller a few days later, who mentioned that the product was being shipped. It has been three weeks and I can't get through to the seller anymore.*"
- **Document C**: "*I have been talking to this girl on WeChat. She advertised her sexual services and asked if I want to meet. When I arrived at the agreed location, a man called me, saying that the girl was under-aged. He threatened me to buy iTunes gift cards or I will be reported to the Police.*"

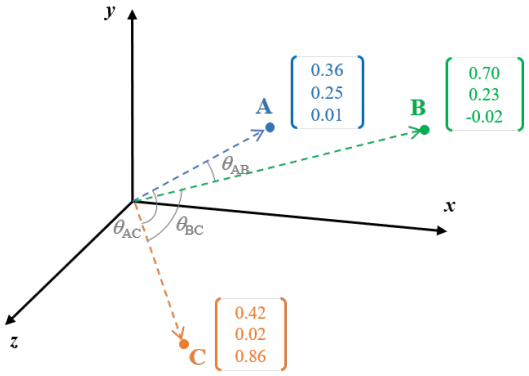


Figure 6.3: Projection of document embeddings on a three-dimensional vector space

By observation, it is apparent that Documents A and B are similar, whereas Document C is the least similar to Documents A and B. Cosine similarity provides a quantitative method of measuring the degree of similarity amongst them. Figure 6.3 shows the mapping of three-dimensional document embeddings representing Documents A, B and C in a vector space. Applying Equation (6.1), the cosine similarity between Documents A and B is computed as $\cos\theta_{\text{AB}} = 0.928$, that between Documents B and C as $\cos\theta_{\text{BC}} = 0.399$, and that between Documents A and C as $\cos\theta_{\text{AC}} = 0.393$. These similarity scores affirm our qualitative observation about their relative similarities. Another characteristic about cosine similarity is that the length of a scam report does not affect its cosine similarity with other reports. In the above example, Document B had high cosine similarity with Document A, even though the former contained longer text, while Documents B and C had low cosine similarity despite being roughly of the same length.

6.2 Methodology

This section describes the methodology for training and evaluating doc2vec models on our text corpus. We used `Gensim` [89], which is an open-source Python library that provides doc2vec implementation amongst other NLP tasks.

6.2.1 Training the Models

In multi-class classification described in Chapter 5, we used cross-validation to validate our models as they trained. Training and validation losses were closely monitored during training and the early stopping technique was applied to avoid over-fitting. For unsupervised algorithms, there is currently no well-researched methodology to determine the optimal amount of training and to identify over-fitting. In the case of doc2vec, `gensim`'s doc2vec implementation does not come with the functionality to monitor losses during training like `keras` or other Python libaries do for deep learning models.

Given these considerations, our methodology was guided by the advice of Mohr [90], the main developer of the doc2vec algorithm in `Gensim`. According to Mohr [90], the optimal amount of training for a doc2vec model can be determined by increasing the number of epochs "until it stops helping downstream evaluations". Instead of partitioning our text corpus into training and test sets as we did in supervised classification, the entire text corpus was used to train doc2vec models. The pre-processed text[1] from all scam reports were first tokenised. Each scam report was assigned a unique document tag ID. We discarded words which occurred only once in the entire text corpus as training such infrequent words could compromise model's performance [91].

Our methodology in training doc2vec models involved a two-staged process. In the first stage, we experimented with training the models for 10, 25, 50 and 100 epochs. We also varied the dimensionality of document embeddings — 20, 30, 40 and 50 dimensions. For each combination, we trained both PV-DM and PV-DBOW algorithms. A total of 32 different models were trained. Based on the results from the first stage, we determined the optimal number of dimensions of document embeddings and the better-performing doc2vec algorithm. In the second stage, we varied the number of epochs of training over a wider range, whilst using the selected doc2vec algorithm and the optimal dimension size. We experimented with the following number of epochs: 25, 50, 75, 100, 150, 200, 250, 300, 400 and 500. Based on the advice of Mohr, we then determined the optimal amount of training by monitoring the number of epochs it took before a model's evaluation performance started to decline.

6.2.2 Evaluating the Models

Unlike the evaluation methodology in multi-class classification of scam reports, there was no test set to evaluate trained doc2vec models. Instead, we adopted two approaches — Self-Similarity Index and Similarity-Dissimilarity Quotient — to assess the quality of document embeddings inferred by each doc2vec model.

[1] This version of pre-processed text was before removing stop words and lemmatisation.

Self-Similarity Index

The *Self-Similarity Index* (SSI) was an idea proposed by Řehůřek [91]. It measures the extent to which a document embedding inferred by a doc2vec model is most similar to itself and not to the embeddings of some other documents. Figure 6.4 illustrates our methodology in computing SSI. First, a document embedding was inferred by one of the doc2vec models for document tag ID #2. Using the inferred document embedding, cosine similarities with every document in our corpus, including document tag ID #2 itself, were computed. Next, the cosine similarity scores were sorted in descending order. Finally, the rank number corresponding to document tag ID #2 was extracted. In this example, the rank number is 2. This process was repeated for all scam reports.

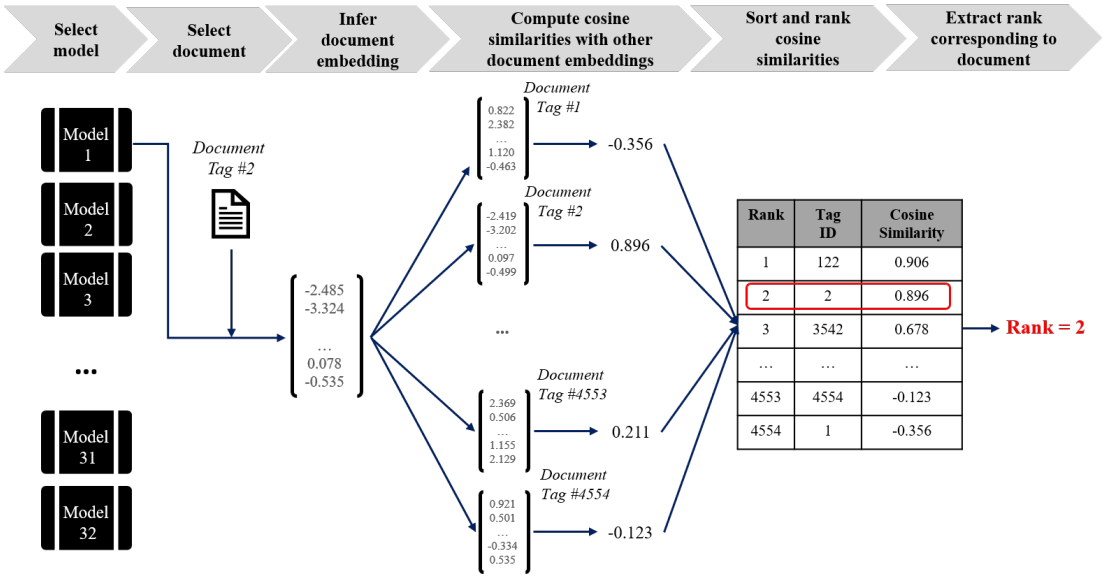


Figure 6.4: An illustration of the process of computing SSI

With 4,554 scam reports in our corpus, there were 4,554 rank numbers. The better a doc2vec model, the more likely it was able to infer document embedding that would be most similar to an embedding of the same document by cosine similarity. In other words, a better doc2vec model would have a higher proportion of '1's amongst the list of rank numbers and thus a higher SSI. The process of computing SSI was then repeated for all 32 trained doc2vec models.

Similarity-Dissimilarity Quotient

Besides being similar to themselves, we believed that document embeddings inferred by a doc2vec model should also be able to recognise similar and dissimilar scam reports. Using the concept of vector semantics, we hypothesised the following: one, document embeddings of similar scam reports are closer in a multi-dimensional space and have high cosine similarity scores; and two, document embeddings of dissimilar scam reports are further apart in a multi-

dimensional space and have low cosine similarity scores. It is with this motivation that we designed a novel framework called the Similarity-Dissimilarity Quotient (SDQ).

Unlike the SSI which compares similarity of a scam report relative to itself, the SDQ compares similarity of a scam report against other scam reports. In designing this framework, we envisaged SDQ to be a single number between 0 and 1. This number would quantify a doc2vec model's ability to produce document embeddings that are capable of recognising similar documents and dissimilar documents in our corpus. In addition, it should have a uni-directional property, that is, the higher the number, the better the model.

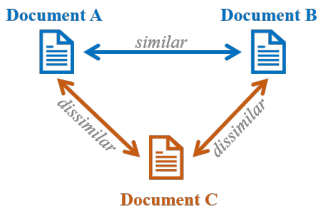


Figure 6.5: An illustration of a triplet consisting of three candidate documents

Consider a set of three candidate documents — Documents A, B and C — shown in Figure 6.5. Documents A and B are similar, but they are both dissimilar to Document C. In this chapter, a set of three candidate documents with these properties will be known as a *triplet*. We have previously introduced the idea of a triplet in Section 6.1.3 when we explained cosine similarity. The SDQ quantifies the ability of a doc2vec model to infer document embeddings, such that Documents A and B have high cosine similarity score, whereas Documents A and C, as well as Documents B and C, have low cosine similarity scores. The first step in evaluating doc2vec models using SDQ involved manually selecting eight triplets from our text corpus. In each triplet, two scam reports were similar and of the same scam type, while the other was dissimilar to the first two and of a different scam type. All eight triplets are presented in Appendix i. Figure 6.6 shows an illustration of how SDQ was computed.

For each triplet, we used one of the trained doc2vec models to infer document embeddings for the three scam reports. Using these document embeddings, we computed cosine similarity scores between one another. Following this, we used the sigmoid function to transform all cosine similarity scores. The sigmoid function was previously defined in Equation (5.3) of Chapter 5. There were two main properties of the sigmoid function that made it suitable for computing SDQ. One, it transforms negative cosine similarity scores to positive values, which are easier to work with when computing SDQ. These positive values then make SDQ more intuitive to interpret. Two, it is a monotonically-increasing function, which means that the order of cosine similarity scores is preserved.

Next, we computed absolute SDQ, which we have defined by the following equation:

$$\mathrm{SDQ}_{\mathrm{abs}} = \frac{\sigma(\cos\theta_{\mathrm{AB}})}{\sigma(\cos\theta_{\mathrm{BC}}) \times \sigma(\cos\theta_{\mathrm{AC}})}, \tag{6.2}$$

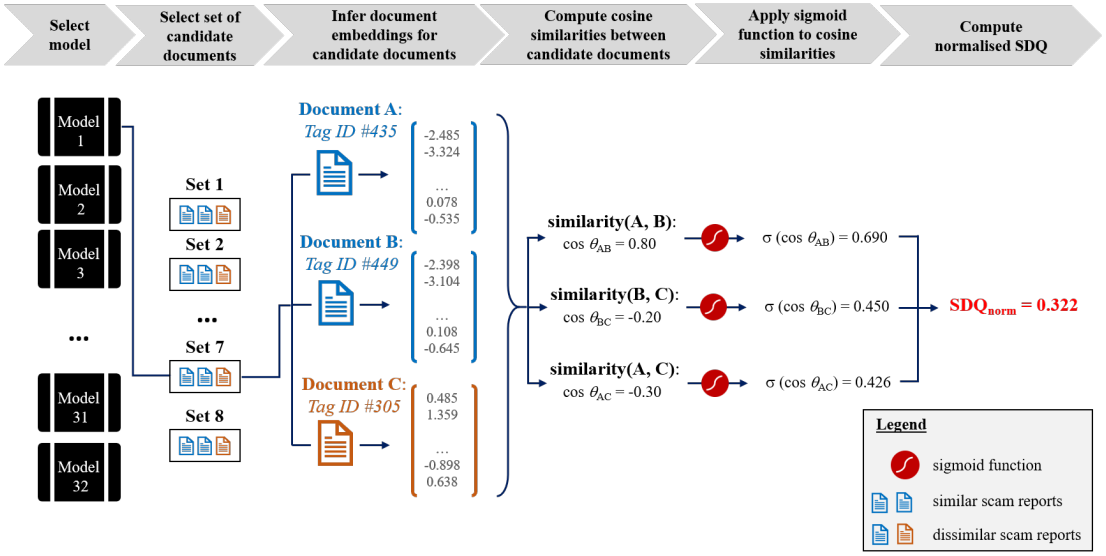


Figure 6.6: An illustration of the process of computing normalised SDQ

where $\cos\theta_{\text{AB}}$ is the cosine similarity between Documents A and B, $\cos\theta_{\text{BC}}$ is the cosine similarity between Documents B and C, and $\cos\theta_{\text{AC}}$ is the cosine similarity between Documents A and C. The more similar Documents A and B are, the bigger the numerator in Equation (6.2) and the higher the absolute SDQ. Conversely, the greater the dissimilarity between Documents B and C, and between Documents A and C, the smaller the denominator and the higher the absolute SDQ.

Absolute SDQ, however, does not fall within the desired range between 0 and 1. Given that $\cos\theta \in [-1, 1]$, the maximum and minimum values of absolute SDQ are

$$\text{SDQ}_{\text{max}} = \frac{\sigma(1)}{\sigma(-1) \times \sigma(-1)} \approx 10.107$$

and

$$\text{SDQ}_{\text{min}} = \frac{\sigma(-1)}{\sigma(1) \times \sigma(1)} \approx 0.503.$$

Therefore, to obtain a metric that ranges between 0 and 1, we normalised the absolute SDQ by the following expression to derive *normalised SDQ*:

$$\text{SDQ}_{\text{norm}} = \frac{\text{SDQ}_{\text{abs}} - \text{SDQ}_{\text{min}}}{\text{SDQ}_{\text{max}} - \text{SDQ}_{\text{min}}}. \tag{6.3}$$

The higher the normalised SDQ, the better a doc2vec model is in inferring document embeddings which are capable of recognising similar and dissimilar documents in a triplet. For each model, the process of computing the normalised SDQ was repeated for all eight triplets before the average normalised SDQ score was taken.

6.3 Results and Discussions

In this section, we present and discuss results of the experiments in two aspects. The first analyses the relative effectiveness of SSI and normalised SDQ in evaluating doc2vec models. This is followed by a comparison of doc2vec models to select the most optimal model.

6.3.1 Assessing Effectiveness of Evaluation Metrics

The focus of this sub-section is to understand the relative behaviours of SSI and normalised SDQ. Figure 6.7 shows results of the doc2vec models being evaluated with both SSI and normalised SDQ. The key observation is that both metrics exhibited different behaviours in capturing performance of the models. Similarity scores in terms of SSI were observed to "hit the ceiling" from 25 epochs of training onwards, regardless of dimension size. This was not the case when normalised SDQ was used. Normalised SDQ continued to increase with the number of epochs, though seemingly at a decreasing rate. The difference in behaviours likely stemmed from the fundamental difference in how each metric measured performance of doc2vec models.

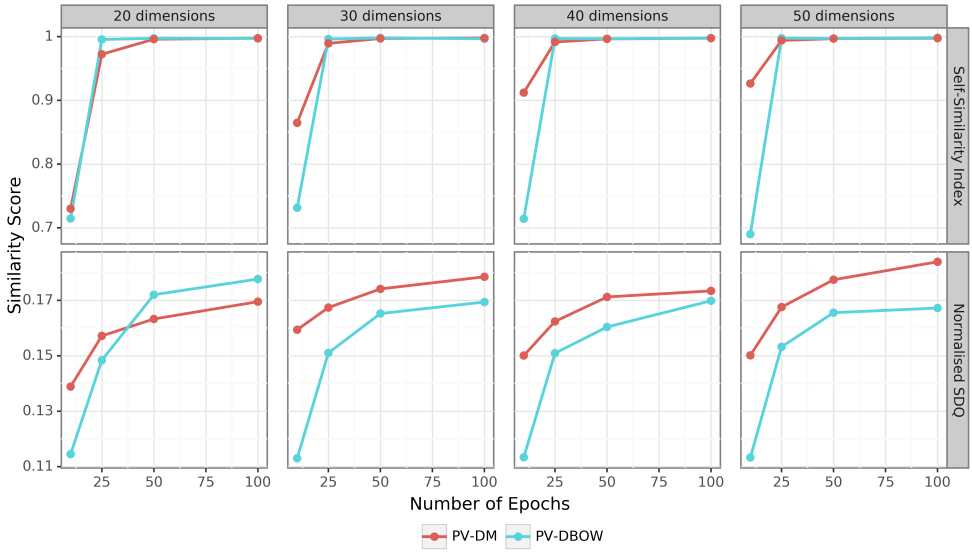


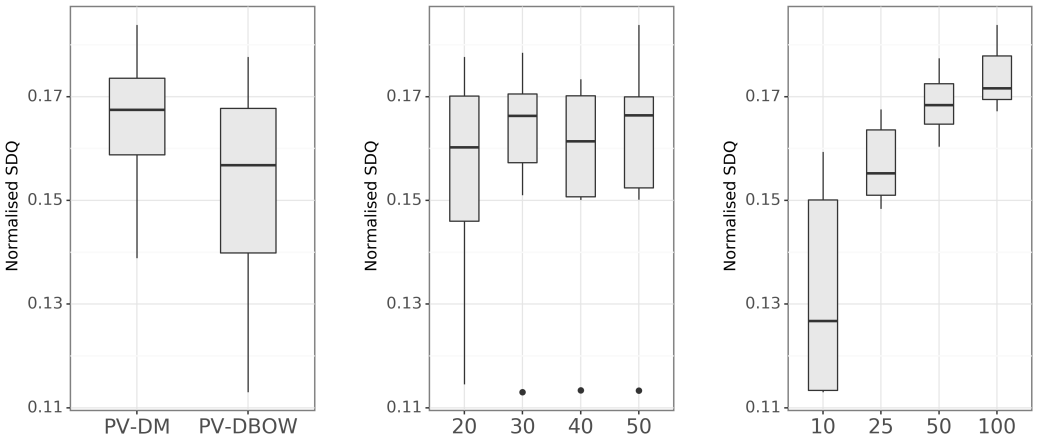
Figure 6.7: Evaluation results of doc2vec models using SSI and normalised SDQ

This observation, on one hand, highlighted the inadequacy of SSI in aiding model selection beyond 25 epochs. Most models trained with 25 epochs or more appeared to achieve almost perfect SSI, making it difficult to select an optimal model. There appeared to be little value in using SSI for model selection, since most models were able to produce document embeddings similar to itself. On the other hand, this observation highlighted the strength of our new framework in distinguishing models. In particular, the fact that normalised SDQ showed continuous increases indicates that as the models trained more, they improved in their abilities to produce document embeddings that recognised similar and dissimilar candidate scam reports within the

triplets. With this in mind, normalised SDQ was assessed to be more effective in aiding our subsequent model selection.

6.3.2 Selecting the Best Model

This sub-section analyses the results of both stages of model training using normalised SDQ with the aim of selecting the most optimal model. In the first stage, 32 doc2vec models were trained with different parameters. Figure 6.8 shows boxplots summarising the results of these doc2vec models. From Figure 6.8a, PV-DM was notably the more superior algorithm, with a median normalised SDQ of 0.167 compared to a median normalised SDQ of 0.157 for the PV-DBOW algorithm. As for dimension size, Figure 6.8b shows that models with document embeddings of 30 and 50 dimensions had median normalised SDQ of 0.166, outperforming those with document embeddings of 20 and 40 dimensions. Given that only a limited number of dimension sizes were experimented, it is inconclusive, at this stage, how dimensionality of document embeddings influences performance of doc2vec models.



(a) Type of doc2Vec model (b) Dimension size (c) Number of epochs

Figure 6.8: Boxplots summarising performance of 32 trained doc2vec models

From Figure 6.8c, there is strong evidence that models that were trained longer achieved higher normalised SDQ. The median normalised SDQ with 10 epochs was 0.127. This increased to 0.155 with 25 epochs, 0.168 with 50 epochs and 0.172 with 100 epochs. This increasing trend had two implications: one, models which were trained longer performed better; and two, there was scope to further improve performance with more than 100 epochs of training.

This leads us to the second stage. In this stage, we varied the number of epochs of training over a wider range, whilst using models with the PV-DM algorithm and document embeddings with 50 dimensions. The results presented in Figure 6.9 show that normalised SDQ improved continuously as the number of epochs increased until 150 epochs. The model with 150 epochs attained the highest normalised SDQ of 0.184. Beyond this point, the performance started to decline, possibly due to over-fitting. Taking into consideration the advice of Mohr [90], it

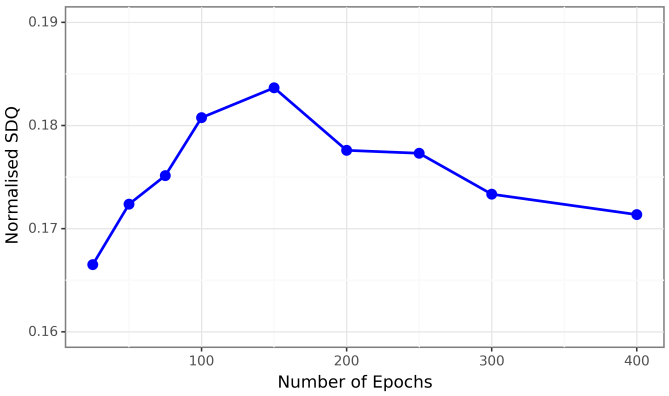


Figure 6.9: Effect of number of epochs on model performance

was assessed that the doc2vec model trained with 150 epochs was the most optimal model for subsequent inferences. This model contained 50-dimensional embeddings not only for each of the 4,554 scam reports, but also for each word in the training data. In addition, it can be used to infer a 50-dimensional embedding for any new scam report.

6.4 Limitations

This section describes two limitations of the work presented in this chapter. The first concerns data quality. The quality of document embeddings inferred by doc2vec was largely dependent on the text used to train the models. Our text data was highly unstructured and inconsistent. Since it was written by different victims, there were huge linguistic variations. For instance, a word could be spelled in numerous ways. These inconsistencies introduced noise in the data which might have hurt model performance. One way to mitigate this was to iteratively identify inconsistencies and rectify them. Although this was done during data cleaning and text pre-processing, it was not feasible to completely eliminate them. Another way was to ignore words which only occurred once in the entire corpus. While this helped to reduce noise in the training data, there were inconsistencies that occurred appeared more than once, like the following: “recived”, “frens”, “ciggarette”, “rrhh”, “s” and “k”. Ignoring words that occur two times or less would also not be ideal since it would reduce the number of words in the training data.

The second limitation relates to our methodology of choosing candidate scam reports for the eight triplets. As much as care was exercised in pre-selecting triplets such that in each triplet, two scam reports were similar and one was dissimilar, it was after all based on subjective judgement. It was, by no means, the best approach. An alternative way of identifying triplets would be to first randomly select n scam reports from the corpus, and then for each scam report use a trained doc2vec model to find the second-most similar as well as the least similar scam reports by cosine similarity. This involves some reverse-engineering since it presumes an already trained doc2vec model. However, a potential advantage is that there is lesser ambiguity for

models in recognising similar and dissimilar scam reports from each triplet given that there was a quantitative basis in selecting them.

6.5 Alternative Approaches

We conceive two alternative approaches that would help meet the research goals of finding similar scam reports and generating key terms The first is to encode scam reports as document embeddings using methods besides doc2vec. One way is to average the word2vec embeddings of words that a scam report contains. Another is to leverage trained deep learning models from Chapter 5. The embedding layers of these models already contained learned weights corresponding to each word in the training data. Therefore, given a new scam report, its embedding can be obtained by averaging the embeddings of its words from the embedding layer.

The second alternative approach is to train doc2vec models on text without stop words. In our work, we did not remove stop words from the text corpus used to train doc2vec models. This was because we assessed that stop words played important roles in the grammatical structures of scam reports and thus would contribute meaningful information towards good document embeddings. Notwithstanding, there is scope to further investigate the quality of document embeddings from doc2vec models trained on a corpus without stop words.

6.6 Summary

On overall, this chapter has explained how our work leveraged the concept of vector semantics to encode scam reports as document embeddings. In particular, we described how doc2vec models, using both PV-DM and PV-DBOW algorithms, were trained and evaluated. In evaluating each doc2vec model, we used two metrics, SSI and normalised SDQ, the latter being a novel framework which examines doc2vec models in terms of the quality of their document embeddings in recognising similar and dissimilar scam reports. Normalised SDQ was assessed to be more effective in aiding model selection. Thereafter, using normalised SDQ, we selected, as the most optimal model, the model that was trained with 150 epochs and 50-dimensional document embeddings using the PV-DM algorithm.

The doc2vec model which we have selected contained 50-dimensional embeddings of all scam reports. There are several use-cases for these document embeddings. In the chapter that follows, we will demonstrate how they can be utilised in two key ways: finding of similar scam reports from our existing text corpus given a new scam report and extracting key terms from a set of similar scam reports. In the process, we seek to validate the concept of vector semantics in terms of whether similar scam reports were indeed close to each other in a multi-dimensional vector space, a hypothesis put forth by Le and Mikolov [88].

Chapter 7

Putting It All Together: A Case Study of the High Court Impersonation Scam

Until now, the focus has been on applying NLP techniques and training machine learning models, both supervised and unsupervised, on our text corpus. In Chapter 5, we selected a LSTM model for multi-class classification of scam reports, and in Chapter 6, a doc2vec model to encode scam reports as document embeddings. The discussions, thus far, had not dwelt on how these individual parts can be drawn together for actionable insights. With this in mind, this chapter takes a holistic approach to show various ways in which our work can be applied in a real-world context. We do so through a case study of the High Court impersonation scam.

In this chapter, we first provide a brief background of the High Court impersonation scam in Singapore. We also give a theoretical overview of TF-IDF and the Jaccard similarity metric, both of which were essential in generating key terms from similar reports. Next, using a hypothetical scam report describing a High Court impersonation scam, we demonstrate the following: (1) how scam reports in the existing text corpus with similar to Document X can be found; (2) how key words and phrases indicative of their modus operandi can be extracted from these similar reports; (3) how Document X can be classified, using both supervised and unsupervised methods. These three areas correspond closely to our research goals set out in Section 1.2.

7.1 Introduction

7.1.1 A Brief Background of the High Court Impersonation Scam

Based on news archives, the High Court impersonation scam first emerged in Singapore sometime in mid-2018. Victims would receive a call purportedly from the "Singapore High Court", with an automated recorded message in English and Mandarin. Victims would be asked to press "9" before being directed to a person claiming to be a Court officer. Victims would be told to attend Court on the pretext that they had been involved in a crime or had pending summons. The

caller would then request for victims' personal details such as names and identification numbers [92]. This scam faded in prevalence before re-emerging sometime around April 2020 [93].

7.1.2 Term Frequency-Inverse Document Frequency

In Chapter 6, we briefly introduced the BOW concept of representing text as numbers and some of its limitations. Another weakness of BOW is that words which appear frequently in documents, such as "a", "the" and "which", are given higher scores, though they carry limited meaning about the document. TF-IDF addresses this limitation of BOW. It consists of two independent terms: *Term Frequency* (TF) and *Inverse Document Frequency* (IDF). TF measures the frequency of a term in a document. The more frequent a term is in a document, the higher the TF score. IDF, on the other hand, rescales the TF score according to how often a term appears across all documents in the corpus.

Mathematically, the TF-IDF score of term i in document j is given by:

$$w_{i,j} = \text{TF}_{i,j} \times \log(\frac{N}{d_i}), \tag{7.1}$$

where $\text{TF}_{i,j}$ is the TF of term i in document j, N is the total number of documents and d_i is the number of documents that contain term i. The latter term on the right-hand side of Equation (7.1) corresponds to IDF of term i. The less frequent a term i appears across all documents in the corpus, that is, the smaller the value of d_i, the higher the TF-IDF score for term i.

One of the strengths of TF-IDF is its effectiveness in discriminating terms which are unique to a set of documents from those which are not. In practice, it brings to the fore words that define meanings of documents. We briefly witnessed this during our EDA in Section 4.2.2, when TF-IDF was used to highlight words that uniquely defined each scam type. For example, unique words generated from credit-for-sex scams using TF-IDF included "girl", "meet", "money" and "transfer". In Section 7.3, we will apply TF-IDF to generate key words and phrases from a set of similar scam reports.

7.1.3 Jaccard Similarity

Besides using cosine similarity, it may sometimes be of benefit to quantify similarity in terms of words using *Jaccard similarity*. In computing Jaccard similarity between two documents, each document is regarded as a set of words. Jaccard similarity measures the proportion of words that two documents have in common. Mathematically, it is defined by

$$\text{Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|}, \tag{7.2}$$

where A and B represents sets of words in the two documents. A Jaccard similarity of 1 implies that two documents are exactly the same and a Jaccard similarity of 0 means otherwise. The

higher the Jaccard similarity, the more words two documents have in common.

7.2 Finding Similar Scam Reports

Our first research question was, "Given textual description of a scam report, which other scam reports share similar modus operandi?" In this section, we demonstrate how this research question can be addressed. We explore three approaches: a vector-based approach using cosine similarity, a text-based approach using Jaccard similarity, and a hybrid approach using both cosine and Jaccard similarities. In the remainder of this chapter, we will perform inferences based on the following hypothetical scam report, which we will refer to as 'Document X'.

> "*I received a scam call. It was an automated voice from the Singapore High Court, stating that I have an outstanding summon. I was asked to pay it.*"

7.2.1 Vector-Based Approach

A vector-based approach refers to using only cosine similarity to measure similarity between document embeddings of scam reports. Before proceeding further, it is useful to gain some intuition on how the concept of vector semantics manifested in document embeddings produced by our selected doc2vec model.

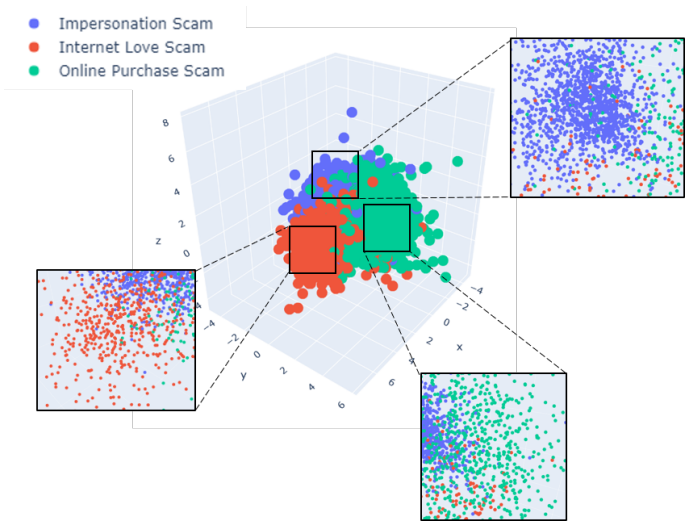


Figure 7.1: Projection of document embeddings onto a 3D vector space

Our selected model contained document embeddings for all scam reports in our corpus. These 50-dimensional embeddings can be extracted from the model and be projected onto a three-dimensional (3D) vector space using a dimensionality-reduction method such as Principal

Component Analysis (PCA)[1]. Figure 7.1 shows the 3D projection of document embeddings of the top three scam types using PCA. This 3D scatter plot was drawn using the Python library, `plotly` [94], with each data point representing a scam report.

What was striking from Figure 7.1 was that embeddings of documents of the same scam type tended to cluster together. This observation validated the vector semantics concept of capturing meanings of text as vectors of numbers. It also supported the hypothesis that similar scam reports were closer in the vector space. Another observation was that there were no clear boundaries between the different clusters. As the insets in Figure 7.1 show, many document embeddings were interspersed within a cluster of a different scam type. This could be due to the complex nature of scams and that scam types are not mutually exclusive. For instance, an internet love scam might entail an element of impersonation and be represented by an embedding positioned between the clusters of internet love scam and impersonation scam.

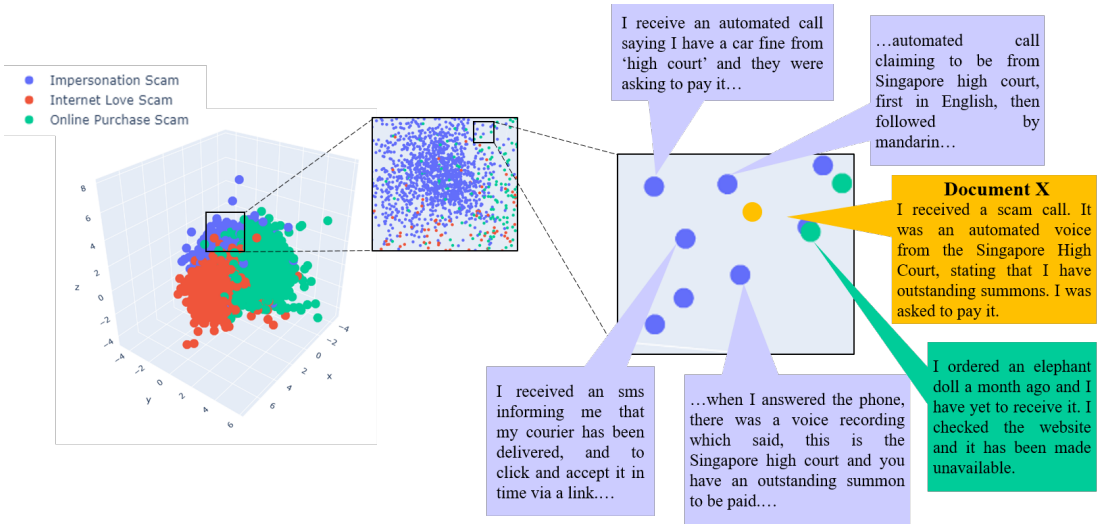


Figure 7.2: An illustration of Document X's embedding on a 3D vector space

Using our doc2vec model, we can infer a 50-dimensional document embedding for Document X. Figure 7.2 shows an inset depicting the position of this document embedding on a 3D vector space, indicated by the data point in orange. The data points in the immediate neighbourhood of the orange data point can be thought of embeddings of scam reports which are most similar to Document X by cosine similarity. In this illustration, several data points close to the orange data point were indeed pertaining to High Court impersonation scams, but there was also others which were unrelated.

Having gained a better appreciation of how vector semantics is a core idea in finding similar scam reports, we now turn our attention to the actual task of finding similar scam reports. Using Document X as the input text, we extracted the top eight most similar scam reports from our text corpus by cosine similarity. These are shown in Table 7.2. Amongst them, only two scam

[1] PCA is a method for reducing dimensionality of high-dimensional data. It transforms a large set of features into a smaller set that retains the most valuable information.

reports, document tag IDs # 972 and #1787, pertained to the High Court impersonation scam. Others were in relation to impersonation of other entities such as the police, DHL and Singtel, a tele-communications company in Singapore.

Table 7.2: Top eight most similar scam reports using the vector-based approach

Rank	Tag ID	Cosine Similarity Score	Scam Report (Pre-Processed)	Scam Type
1	972	0.669	call from automated voice message stating that he is from the high court saying that i have missed submitting an important document and asked to press to ask more questions.	Impersonation Scam
2	4339	0.660	i simply ignore and hung up the call	Impersonation Scam
3	1219	0.634	the dhl scam is back received a call that started off with an automated voice message in mandarin, informing me that i have a parcel from dhl. i was asked to press to speak to an operator. hung up the call as i knew it was a scam. call came from this number which might be a spoofed number	Impersonation Scam
4	55	0.627	i received a scam spam call from this number asking for details about my singtel internet connection. i am not even a subscriber of any singnet singtel services. so, i ended the call. i thought it would be great to share my experience here to warn others of this scam.	Phishing Scam
5	2484	0.623	i received a call from on th may. it was a female voice claiming that it was a call from singapore police headquarters. i hung up immediately as i sensed something was wrong.	Impersonation Scam
6	1787	0.621	got a call aug at. private message from this no advising i have outstanding summons not cleared	Impersonation Scam
7	2422	0.619	received this scam call from the singapore police force this morning.	Impersonation Scam
8	825	0.601	i received a call from an unknown number. heard an automated voice message informing me that i have an unclaimed package. it spoke in english, then mandarin. i hung up immediately	Impersonation Scam

7.2.2 Text-Based Approach

As Table 7.2 shows, the scam reports found by the vector-based approach were not entirely of similar modus operandi as Document X's. An alternative approach is using Jaccard similarity. While cosine similarity operates in the vector space, Jaccard similarity deals with the 'text

space'. In this text-based approach, similarity between scam reports is measured based on the proportion of words two scam reports have in common.

In our work, we applied Jaccard similarity with a slight twist. Jaccard similarity, in the conventional way, considers all words in a pair of documents including stop words. Since stop words appear commonly in documents, Jaccard similarity will not provide an accurate picture of the degree of similarity between scam reports in terms of their modus operandi. We postulate that scam reports with similar modus operandi will likely make reference to similar phrases such as names of government organisations, banks and social media platforms. In view of this, we modified the computation of Jaccard similarity to be based on noun phrases[2] in a pair of documents instead of all words. Put differently, the more noun phrases two scam reports have in common, the more similar their modus operandi will be.

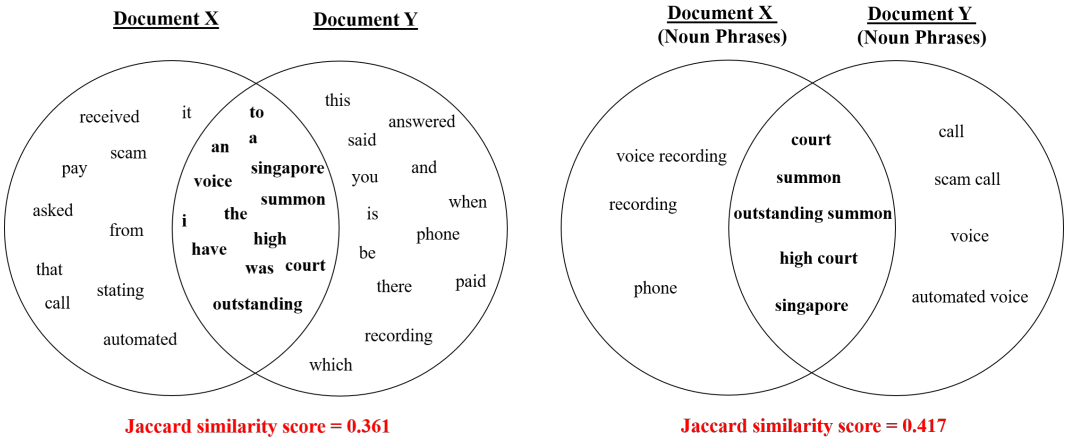


Figure 7.3: Comparison of Jaccard similarity using all words and only noun phrases

To contextualise this idea, consider the example shown in Figure 7.3, where Document X is compared with another document, 'Document Y'. The Jaccard similarity score using noun phrases is different from that using all words. In this case, Documents X and Y were more similar in their modus operandi when compared using noun phrases. We used `spaCy` to identify and extract noun phrases[3] from scam reports. We then computed Jaccard similarity scores between Document X and each scam report in the text corpus in terms of how many noun phrases they had in common. The top eight scam reports most similar to Document X by Jaccard similarity are presented in Table 7.3. Evidently, all eight scam reports were highly similar to Document X in terms of the modality of the High Court impersonation scam, with several noun phrases in common. This demonstrates the effectiveness of using Jaccard similarity with noun phrases to find scam reports with similar modus operandi.

[2] A noun phrase is a phrase containing a noun and optionally other kinds of words such as pronouns and adjectives. Examples include "scam call", "internet connection", "my connection", "local landline" and "police".

[3] Noun phrases were extracted based on two matching patterns: phrases that contain consecutive nouns with at least one noun; and phrases that start with one adjective followed by consecutive nouns with at least one noun.

Table 7.3: Top eight most similar scam reports using the text-based approach

Rank	Tag ID	Jaccard Similarity Score	Scam Report (Pre-Processed)	Scam Type
1457	1866	0.500	automated robot voice called me and said it is from singapore high court, and i have outstanding summon also repeated in mandarin then i hung up. not sure what is the intention of the call.	Impersonation Scam
1175	356	0.462	received a call from at. am on april. it was an automated voice stating the call is from singapore high court. put down the phone immediately after, so did not hear the rest of the message.	Impersonation Scam
490	279	0.400	i got the above call and it says i got an outstanding summon and to dial nine to talk to someone. someone answered and says its singapore high court the person sound like a local singaporean.	Impersonation Scam
501	141	0.400	got a call from a local singapore number. answered the call and it was an automated voice in english saying the high court of singapore was serving a summons on me. i immediately hung up. this is a total scam, the high court of singapore will never serve summons like this.	Impersonation Scam
363	1874	0.385	received a phone call with a voice message saying this is singapore high court. you have an outstanding summon. press.	Impersonation Scam
422	1803	0.385	call with automated voice saying i have summons from singapore high court. press to proceed. someone picked up the call but then did not speak anything.	Impersonation Scam
384	1026	0.375	i received a call from initially it was automated voice saying you have been summoned by singapore high court to get a document, for more details press. as i just received a dhl scam call yesterday, i did not proceed any further.	Impersonation Scam
1757	1840	0.375	received a call from this morning. a computer voice said this is the singapore high court. you have a summon pending. please press after the beep for more information. i became suspicious and cut the call. beware people	Impersonation Scam

7.2.3 Hybrid Approach

Through trial and error, we discovered that the text-based approach tended to be more effective than the vector-based approach for scams which were predictable and had several common characteristics. Our present case study is one such example. The characteristics which were

common across similar High Court impersonation scams included the high court, an automated voice and outstanding summons. For such scams, phrases used by victims to describe their experiences were more likely to be similar.

The text-based approach, however, appeared to work less effectively for scams whose modus operandi were more varied. An example was internet love scams, where scammers used different names on different online dating platforms, claiming to be from different countries and telling different stories. There was a greater variety in the phrases used by victims to describe such scams, which meant that it would be of lesser utility to measure similarity in terms of common characteristics. Instead, measuring contextual similarity between such scam reports using the vector-based approach might be a better choice in identifying similar scam reports.

Notwithstanding, this observation was, at best, subjective and empirical. There may be exceptions. With this in mind, we believe that the method to finding similar scam reports that generalises to any scam reports is a hybrid of both vector-based and text-based approaches. In this hybrid approach, we regard both contextual and characteristic similarities as important, whilst acknowledging that their relative importance will vary depending on the type of scam and our objectives. For example, for Document X, the hybrid approach accorded greater importance to the text-based approach given the predictability of such scams and the effectiveness of this approach. Ultimately, we believe that it is through experimentation that the best way of retrieving scam reports with similar modus operandi can be found.

7.3 Generating Key Terms from Similar Scam Reports

In the previous section, we described various ways of finding similar reports given a scam report. In this section, we take one step further to generate words and phrases which are unique to a set of similar scam reports. In doing so, we aim to answer our second research question, "Given textual descriptions of a set of similar scam reports, what are the common characteristics of their modus operandi?"

Words and phrases which we extract from scam reports will be broadly referred to as *n-grams*, which means a consecutive sequence of n words. An n-gram with one word is a *unigram*, with two words a *bigram* and with three words a *trigram*. We also use the word *terms* interchangeably with n-grams. Given that our text corpus primarily contained victims' accounts of scams, key terms from similar scam reports provided useful information about how scams happened from victims' perspectives.

The first step involved removing stop words from textual descriptions in a set of similar scam reports using the `nltk` library. Next, we applied TF-IDF on a set of similar scam reports without stop words to identify the unique terms. Each term was given a TF-IDF score according to its importance to a given set of scam reports. The higher the TF-IDF score, the more important the term was. Amongst the top 0.5% of scam reports most similar to Document X, we extracted and ranked a total of 150 unigrams, 288 bigrams and 326 trigrams. The top 10 unigrams, bigrams

and trigrams alongside their TF-IDF scores are shown in Table 7.4. It is clear from Table 7.4 that TF-IDF generated n-grams which were unique to the High Court impersonation scam.

Table 7.4: Top 10 n-grams and their respective TF-IDF scores

Unigrams	**TF-IDF Score**	**Bigrams**	**TF-IDF Score**	**Trigrams**	**TF-IDF Score**
call	3.757	singapore high	2.132	singapore high court	2.002
singapore	2.804	high court	2.107	court outstanding summon	1.442
court	2.739	outstanding summon	1.612	high court outstanding	1.442
received	2.731	phone call	1.488	received phone call	1.371
high	2.688	court outstanding	1.468	outstanding summon press	1.319
press	2.617	received phone	1.410	call voice message	1.257
voice	2.567	message saying	1.370	message saying singapore	1.257
saying	2.278	summon press	1.353	phone call voice	1.257
summon	2.184	call voice	1.286	saying singapore high	1.257
outstanding	2.112	saying singapore	1.286	voice message saying	1.257

While Table 7.4 highlights key terms which were characteristic of the High Court impersonation scam, it is less informative about how this scam typically unfolded. To improve on the way such key terms are presented, we further harnessed information in the set of similar scam reports by taking note of the index position corresponding to a particular n-gram in each scam report. For example, the index position of the bigram "singapore high" in document tag ID #1866 was 52, meaning that "singapore high" appeared at the 52nd index position of document tag ID #1866. The n-grams were then arranged by their median index positions across the set of similar scam reports. Table 7.5 illustrates this idea for the top 20 unigrams amongst the top 0.5% of scam reports most similar to Document X.

Table 7.5: Top 20 unigrams arranged by median index positions

Unigrams	**TF-IDF Score**	**Median Index Position**
received	2.731	0.0
automated	1.732	9.0
phone	2.076	9.0
call	3.757	11.5
number	1.281	14.0
voice	2.567	26.5
message	2.073	30.5
saying	2.278	35.0
singapore	2.804	41.0
high	2.688	52.5
court	2.739	57.5
english	1.253	60.5
outstanding	2.112	63.0
summon	2.184	74.5

summons	1.228	78.0
press	2.617	84.5
scammer	1.398	90.0
details	1.727	98.0
chinese	1.106	120.0
hung	1.230	122.0

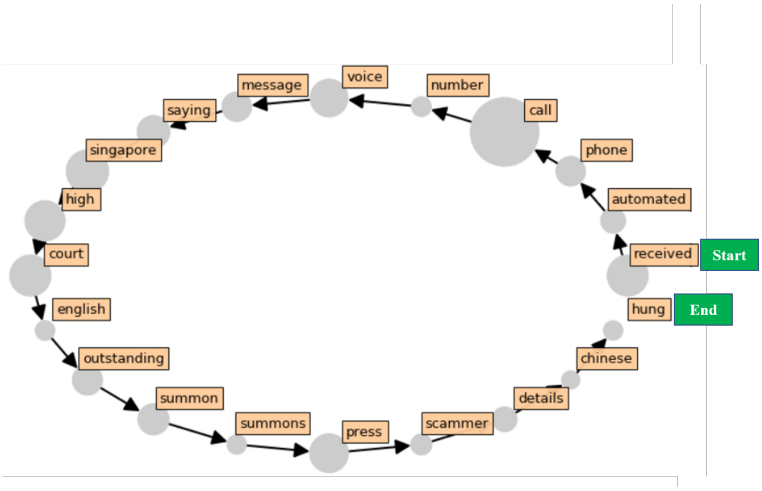


Figure 7.4: A directed graph showing the top 20 unigrams in sequence

When arranged by median index positions in Table 7.5, the sequence of unigrams provided a better idea about how the scam typically took place. The same information presented in Table 7.5 can be visualised as a directed graph, as shown in Figure 7.4. This directed graph consists of a series of nodes, where each node represents an n-gram and the size of the node reflects its TF-IDF score. The directionality of the graph conveys the sequence in which the scams typically unfolded. Besides generating only one type of n-gram from a set of similar reports, we can do so for a combination of unigrams, bigrams or trigrams. Figure 7.5 shows a directed graph depicting a combination of top 15 unigrams and bigrams from the top 0.5% of scam reports most similar to Document X.

To sum up, extracting key terms using TF-IDF enabled us to identify common characteristics of the High Court impersonation scam. Moreover, by arranging them using their median index positions and visualising them using a directed graph, we were able to derive some chronological intuition about the scam. Specifically, it enhanced our understanding of the sequence of events typically involved in the High Court impersonation scam, at least from the victims' points of view. More generally, such insights can allow us to derive patterns in various types of scams and thereby identify potential points of intervention where scams can be disrupted.

Our work presented in this section harnessed information not only from Document X, but also from the collective wisdom of scam reports which were most similar to Document X. The same methodology can be applied to any other types of scams. We envision our work to pave

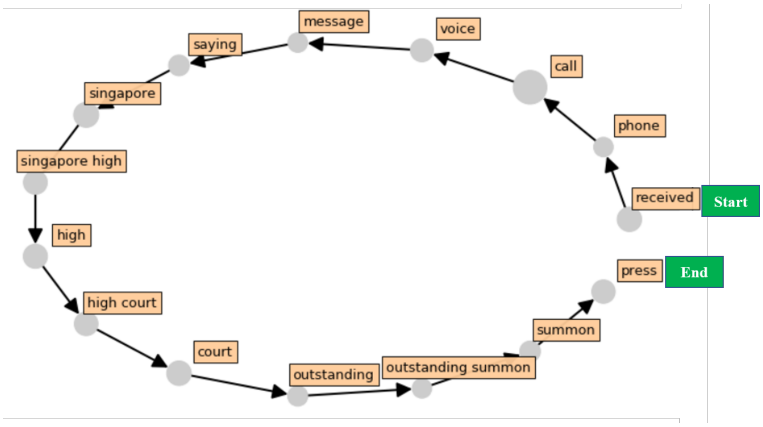


Figure 7.5: A directed graph showing the top 15 unigrams and bigrams in sequence

the way for future research in generating key terms from reports that would aid in script analysis of scams or crimes.

7.4 Classifying Scam Reports

Our final research question was, "Given textual description of a scam report, which category of scam types does it belong to?". In Chapter 5, we sought to address this using deep learning algorithms. The LSTM model that was selected in Chapter 5 can be used to predict the classification of an unseen scam report. In addition, in Chapter 6, we selected a doc2vec model which can generate a 50-dimensional document embedding given an unseen scam report. Besides finding similar scam reports and generating key terms, such document embeddings can also be utilised in classifying a scam report. In this section, we will demonstrate these methods using Document X as the input scam report.

7.4.1 Using the Selected Doc2Vec Model

One way of classifying Document X was using the idea of the *k*-Nearest Neighbours (KNN) algorithm. KNN is a classification algorithm that uses labelled points in a vector space to classify new unlabelled points [61]. It is based on the paradigm that similar data points with the same class labels are in close proximity by Euclidean distance to one another in the vector space [61]. In our case, instead of Euclidean distance, we used cosine and Jaccard similarities to identify reports which were similar to an unseen scam report.

For Document X, we established, in Section 7.2.3, that Jaccard similarity was the more effective way to find similar scam reports. Hence, we can use the data points which represent these similar reports by Jaccard similarity, also referred to as *neighbours*, to classify Document X. Specifically, the predicted classification of Document X is the most common scam type amongst its neighbours. To put this into context, consider a simple illustration in Figure 7.6

which shows a two-dimensional projection of Document X and its neighbours. The proportion of neighbours corresponding to the majority class varies with the value of k. When $k = 4$, 75% of the neighbours were impersonation scams. This proportion changed to 71.4% when $k = 14$.

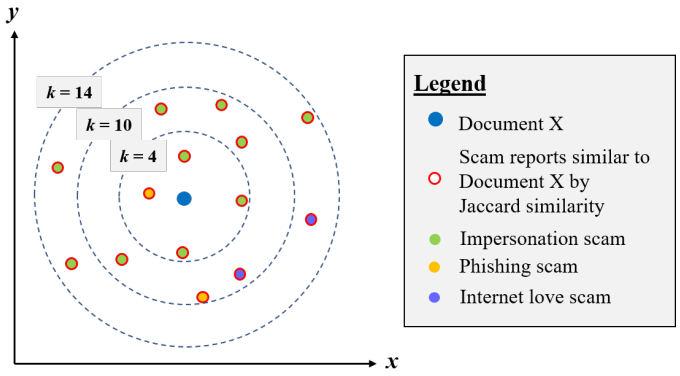


Figure 7.6: An illustration of KNN by Jaccard similarity (k = 14)

Table 7.6: Class distribution of nearest neighbours of Document X for different k values

Percentile of most similar scam reports	**k**	**Classifications of neighbours**	**Percentage of neighbours with corresponding classification**	**Predicted classification of Document X**
0.5%	24	Impersonation Scam	**100.0%**	Impersonation Scam
1%	52	Impersonation Scam	**98.1%**	Impersonation Scam
		Cyber Extortion Scam	1.9%	
2.5%	116	Impersonation Scam	**94.8%**	Impersonation Scam
		Cyber Extortion Scam	2.6%	
		Phishing Scam	1.7%	
		Money Mule Scam	0.9%	
5%	233	Impersonation Scam	**96.1%**	Impersonation Scam
		Cyber Extortion Scam	1.7%	
		Phishing Scam	1.3%	
		Money Mule Scam	0.9%	

The actual distribution of scam types of the k nearest neighbours to Document X for different values of k is shown in Table 7.6. Amongst the top 0.5% of most similar scam reports, all 24 nearest neighbours were impersonation scams. As we examined more similar scam reports, that is, as k increased, other scam types surfaced and the proportion of neighbours corresponding to the majority class changed. In this case, however, varying k had little effect on the proportion of neighbours which were impersonation scams. This gives confidence on the predicted classification of Document X being 'Impersonation Scam'. In practice though, this may not always be true. Additionally, further research and experimentation are necessary to determine the optimal value of k.

7.4.2 Using the Selected Multi-Class Classification Model

An alternative to classifying scam reports is using our trained deep learning models. In Chapter 5, we selected the LSTM model that was trained without pre-trained GloVe word embeddings on a dataset balanced by text augmentation. Each layer of this model contained weights which had been optimised by back-propagation using the training data. It was these weights in each layer of the model that enabled us to predict the classification of a new scam report.

Before making inferences with our trained LSTM model, the text sequence of Document X first needed to be in the correct format. The first step involved pre-processing the sequence, using the same pipeline explained in Section 3.5. Following this, stop words were removed and the text sequence was lemmatised. Next, the pre-processed sequence was tokenised and each token converted to a number corresponding to the same index as assigned during training. The final step was to pad the sequence of numbers with additional '0's to obtain a sequence that was 66-tokens long. This was so as to be consistent with the length of input sequences for models trained on the text-augmented dataset. These steps are summarised in Table 7.7.

Table 7.7: Steps in preparing Document X for inference with trained LSTM model

Steps	**Representation of Document X**
Input text	"I received a scam call. It was an automated voice from the Singapore High Court, stating that I have an outstanding summon. I was asked to pay it."
Text pre-processing	"i received a scam call. it was an automated voice from the singapore high court, stating that i have an outstanding summon. i was asked to pay it."
Remove stop words	"received scam call automated voice singapore high court stating outstanding summon asked pay"
Lemmatisation	"receive scam call automate voice singapore high court state outstanding summon ask pay"
Convert text to sequence	[17, 13, 1, 168, 158, 18, 264, 418, 99, 849, 928, 2, 5]
Pad or truncate sequence	[17, 13, 1, 168, 158, 18, 264, 418, 99, 849, 928, 2, 5, 0, 0, 0, 0, . . . , 0, 0, 0]

With the text sequence of Document X into the correct format, it was fed into the LSTM model to generate a prediction. The final layer of our LSTM model, which was the dense layer, produced six output values. These values were transformed using the softmax function into probabilities that summed to one. The generated probabilities for classification of Document X is reported in Table 7.8. In this case, the predicted classification was 'Impersonation Scam', which was the class that corresponded to the highest probability.

Table 7.8: Predicted softmax probabilities for Document X

Classification	**Predicted probabilities after softmax**
Credit-for-Sex Scam	0.00270

Home/Room Rental Scam	0.00179
Impersonation Scam	**0.98692**
Internet Love Scam	0.00180
Investment Scam	0.00094
Online Purchase Scam	0.00587

7.5 Summary

In essense, this chapter has provided a flavour of how our work can address the research questions. We started off by finding scam reports in our text corpus which were similar to Document X. We explored the vector-based, text-based and hybrid approaches, and concluded that the hybrid approach was the most recommended approach in the interest of generalising to all scam reports. Next, using TF-IDF, we showed how key words and phrases were extracted from the set of scam reports most similar to Document X. These key words and phrases were highly characteristic of High Court impersonation scams. Visualising them in sequence provided a stronger intuition about how these scams typically take place. Lastly, we predicted the classification of Document X in two ways: unsupervised method using the idea behind the KNN algorithm; and supervised method using the trained LSTM model. In this case, both methods generated the same prediction of Document X — impersonation scam. With these tools now at our disposal, we can generate similar insights from any scam reports in free text. In the next chapter, we will further discuss the significance of our work in the broader context of how they can be applied in the fight against scams.

Chapter 8

Significance of Our Work

In the previous chapter, we described various ways in which our research work can be applied to perform various tasks. These include finding scam reports with similar modus operandi, generating key terms from similar scam reports and classifying scam reports. In this chapter, we take a broader view and justify the significance of our work by putting into context how they can translate to practical outcomes in the fight against scams.

8.1 The Need for a Better Understanding of Scams

Scams have become increasingly prevalent in recent years. The predominant use of the Internet to propagate scams means that they transcend geographical boundaries, making any person around the world a likely victim. The faceless nature of many scams today adds to the difficulties faced by law enforcement authorities in identifying perpetrators and bringing them to justice. Moreover, scams impact victims in multitude of ways. Despite constant efforts by authorities in Singapore, the number of scam cases continued to rise.

This formed the backdrop of our research, the motivation of which was to develop tools using machine learning and NLP methods that aid in tackling scams. Our research was premised on the Routine Activity Theory in the context of scams. We postulated that like crimes, the three necessary elements of scams are a suitable target, a motivated offender and the absence of a capable guardian, though they need not converge in time and space as in physical crimes. This theory shaped our understanding of why scams continue to be successful.

It is only by understanding how scams operate and why they are successful that we will be able to devise ways to curb scams. Our research was fundamentally an endeavour towards an increased understanding of scams. With a better understanding, more robust policies and legislation can be enacted. More strategic scam prevention measures can be put in place. More data-driven, targeted public outreach and engagement efforts can be implemented. We envision our work to contribute alongside existing slew of anti-scam efforts in Singapore in two ways: a smarter law enforcement and a more discerning public.

8.2 A Smarter Law Enforcement

The tools we have built serve to enhance the sense-making and analytical capabilities of the SPF, the law enforcement authority in Singapore. The finding of similar scam reports is envisaged to be of benefit, particularly in analysing scam patterns. Scams with similar modus operandi can be linked together and be possibly traced to the same perpetrators or syndicates. This facilitates thorough police investigations and prosecution against perpetrators, thereby strengthening the tough stance against scams. By analysing document embeddings of new scam reports and evaluating their similarities to scam reports in the existing database, the SPF can potentially detect new variants of scams and better understand, in real time, how scams are evolving. This allows the SPF to publicise emergence of new scams early, preventing more victims from falling prey.

Our work in generating key terms provides an efficient method to identify key characteristics of scams and understand the sequence of events involved. One area of benefit is in the script analysis of scams, especially given that script analysis typically involves a resource-intensive process of examining huge amounts of text documents. Moreover, studying victims' scripts, alongside offenders' scripts if they are available, can potentially allow the SPF to identify pinch points in the occurrences of scams. These pinch points can enable more tailored intervention measures. For example, if the name of a money remittance company shows up as a key term from a set of similar scam reports, the SPF can engage the company and its employees to look out for possible scam victims using their company to remit money.

The current process of manually classifying scam reports submitted by victims on 'Scam Alert' is problematic. It requires knowledge, experience and subjective judgement, which means that different annotators may perceive and classify a scam report differently. It is also prone to human error, as we have seen from the substantial number of misclassified reports. This exactly justifies the value for an automatic classifier. We envisage significant utility in alleviating resources required for this task as well as achieving more consistent classifications of scam reports. The role of the human annotator is then transformed to one who provides a secondary layer of check on predictions made by the automatic classifier. Accordingly, more resources can be strategically channelled to enhance anti-scam efforts, such as scam prevention initiatives and outreach efforts.

8.3 A More Discerning Public

Besides enabling a smarter law enforcement, the tools that we have built can also help nurture a more informed public. Searching for other scam reports on a public-facing platform like 'Scam Alert' can allow the public to verify their suspicions before they commit to any deals or respond to requests from scammers. Key terms generated from reports similar to their scam experiences can educate the public on the common tell-tale signs. A classifier that automatically predicts a scam type given a textual description can allow the public to understand the type of scams they

are dealing with and be recommended relevant advice against falling prey.

Collectively, these tools can help to raise general public awareness about scams as well as precautions they should take to safeguard themselves. A more discerning public also means better guardianship and a stronger sense of vigilance within the society, which are important elements that can help nip scams in the bud. After all, no matter how motivated and sophisticated scammers are, a discerning public will always be the strongest defence against scams.

8.4 Summary

Beyond scams, the methodologies that we have used in our research are reproducible for any text data, including crime reports. Similar tools can be built in the context of crimes. Finding of similar crime reports will be hugely beneficial in analysing crime patterns and linking similar crimes. Generating key terms from similar crime reports will aid a great deal in crime scripting and formulation of crime prevention measures. Automatic classification of crime reports will free up resources, enabling law enforcement authorities to focus on other strategic priorities.

Our work in using machine learning and NLP methods is envisioned to play a significant role in the fight against scams. The tools that we have built sharpen the analytical capabilities of law enforcement authorities in better understanding scams and the ways they operate. With such insights, public education and engagement efforts can be more targeted and effective. They also boost the quality of investigations, which in turn serves as a deterrent and reinforces the tough stance against scams. Additionally, they help to create a more vigilant and discerning public, fortifying the society's defence against scams. In the final chapter that follows, we will take stock of the work that we have done and propose future directions.

Chapter 9

Conclusions

9.1 Beyond Our Work: Areas of Future Research

Given that our research dabbled with several novel applications of machine learning and NLP on free text in the domain of scams, and in so doing, addressed gaps in research, we anticipate our work to be a starting point for further research as well as real-world applications for law enforcement authorities in tackling scams. In each of Chapters 5 and 6, we outlined several alternative approaches that could help achieve similar outcomes in terms of our research goals. In this section, however, we explore different trajectories of this research that we envisage can also play significant roles towards tackling scams. In the paragraphs that follow, we highlight three areas of future research. These are also reflected in an updated framework shown in Figure 9.1.

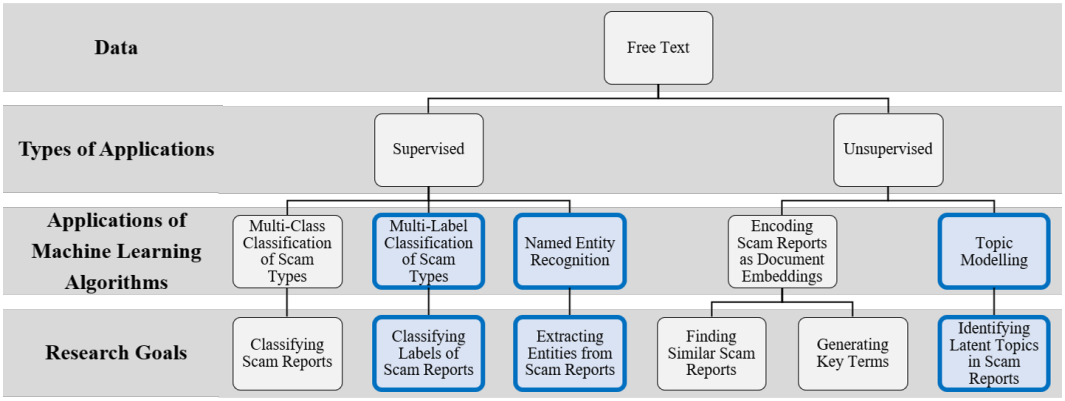


Figure 9.1: Updated research framework with potential areas of future research

One area of future research is multi-label classification of scam reports. One drawback about multi-class classification is that it involves a "massive loss of information" [23]. It assumes that scam types are mutually exclusive and that each scam report can belong to only one scam type. In practice, this is not always true especially given the complex nature of some scams. Unlike

multi-class classification, multi-label classification classifies scam reports by labels instead of discrete scam types. These labels are pre-defined topics or tags associated with scam reports. In multi-label classification, a scam report can be assigned multiple labels. Given an unseen scam report, a trained multi-label classifier will predict one or more labels associated with the report. Therefore, multi-label classification can reveal more information about a scam report than multi-class classification, which only predicts the scam type.

Since multi-label classification is a supervised task, scam reports first need to be annotated with labels before they can be trained with machine learning algorithms. However, manual annotation is a time-consuming process. An alternative to this is topic modelling, which is another area of future research. Topic modelling is an unsupervised task of identifying key topics relevant to a set of documents [21]. For instance, Kuang *et al.* [23] used topic modelling to discover latent topics from police narrative text about crime events. In a similar way, we can explore various topic modelling techniques such as Latent Semantic Analysis and Latent Dirichlet Allocation to extract hidden topics from scam reports. Like the work of Birks *et al.* [24], clustering such topics can potentially provide deeper insights on the intricacies within each scam type.

A third possible area of future work is Named Entity Recognition. Scam reports in our text corpus contained many different named entities, such as names of people, organisations, geographical locations and social media platforms. These entities and the relations between them hold a wealth of insights, which can be exploited in different ways. Al-Zaidy *et al.* [16] and Elyezjy and Elhaless [17] used extracted entities to construct crime networks. Similarly, entities extracted from our scam reports can be used to build a knowledge graph, which is a powerful information mining tool. For instance, given a new scam report and its named entities, a knowledge graph can inform us not only of other similar scam reports containing those entities, but also on relations between the entities. This can be an added tool in analysing modus operandi of scams.

9.2 A Recap of Our Research Questions

In this dissertation, we set out to apply supervised and unsupervised NLP methods on free text in scam reports towards tackling scams. We began by conceptualising three research questions and sought to answer them based on our proposed framework shown in Figure 1.1.

The first research question was, "Given textual description of a scam report, which other scam reports share similar modus operandi?" To answer this question, we leveraged the idea of vector semantics to represent scam reports as document embeddings and explored both vector-based and text-based approaches of finding similar scam reports. We established that both contextual and characteristic similarities were important and concluded that the hybrid approach would be most generalisable to any scam reports.

Next, given a set of similar scam reports, we showed how TF-IDF was used to extract unique

words and phrases which were indicative of the modus operandi of similar scam reports. This addressed our second research question, "Given textual descriptions of a set of similar scam reports, what are the common characteristics of their modus operandi?"

The final research question was, "Given textual description of a scam report, which category of scam types does it belong to?". We approached this in two ways: using a LSTM model that was trained without GloVe embeddings on a dataset balanced by text augmentation; and using 50-dimensional document embeddings from the doc2vec model trained with the PV-DM algorithm for 150 epochs.

We have shown, through the case study of the High Court impersonation scam, how these questions can be answered using findings from our research and tools that we have built. Furthermore, as alluded to in Chapter 8, we envisage our work to play significant roles in helping to inoculate the society against scams, through a smarter law enforcement and a more discerning public.

9.3 Final Words

We began our research with a blank slate, sourced our own data, designed various algorithms and now have a variety of tools for analysing and understanding scams. More than anything, our work underscored the tremendous potential of machine learning and NLP methods in unlocking and harnessing hidden insights from free text. We envision our work to pave the way for further research into more cutting-edge applications of machine learning and NLP on free text towards tackling scams. The work that we have started towards fighting scams does not end here. With new state-of-the-art techniques being developed rapidly, the possibilities are endless. We are just getting started.

Appendix A: Descriptions of Scam Types

Table 1: Description of scam types (Source: 'Scam Alert')

Scam Type	**Description**
Apple Scam	Scammers tell victims that they or their loved ones are about to experience misfortune, which they can prevent by performing a ritual. As part of the ritual, money or valuables are placed in a plastic bag or container. Victims are told that the bag or container can only be opened a few days later or the ritual will not work. When the proper time has passed, victims open the bag to find that the valuables have been replaced by worthless items like fruit, newspapers or sugar.
Car Rental Scam	Scammers trick victims into paying a deposit or the full rental fee before receiving the car. After payment has been made, victims find that the agency and car do not exist.
Cold Call Supplier Scam	This scam targets owners of trading firms. It usually begins with a call from a person overseas, who offers the company owner exclusive rights to sell a product in Singapore. After they accept the offer, the trading firm owner receives a large order for the product, which actually comes from the scammers. To fulfil those orders, the victim orders and pays for the items. The victim soon realises that the items will never arrive and the scammers have disappeared.
Credit-for-Sex Scam	In this scam, a stranger befriends her victim through social media platforms such as WeChat. The scammer talks the victim into buying them a purchase or gift card (e.g. Alipay Purchase Cards, iTunes cards, etc) in exchange for a meet-up, date or sexual favours.
Cyber Extortion Scam	Scammers befriend victims online and then coax them into performing an indecent act on camera. Afterwards, the scammers use the video footage or images of the act to extort money from the victims.
Home/Room Rental Scam	Scammers offer a room or house for rent and use high-pressure tactics to get victims to pay the rent in advance. In these cases, the scammers are not authorised to rent out the property or the property may not even exist. In another variation of this scam, the scammers pretend to be interested foreign tenants looking to rent a place. They will claim to have made payment for the deposit, after which the victims will receive fake emails from PayPal asking for a fee before the payment can be released to them. In some cases, the scammers may ask the landlords to help them pay their movers' fees.

Table 1 continued from previous page

Scam Type	Description
Impersonation Scam	There are several variations of this scam. One involves a phone call from someone purporting to be a government official, such as a police officer, immigration officer or court official. In another variation, the caller might claim to be an employee or representative of a Chinese bank or courier company. The caller might claim that your identity was used to send parcels containing fake passports or weapons, or to apply for overseas credit cards. They then refer you to another caller claiming to be a Chinese official, who will ask — or even threaten — you to give them personal information such as your passport or bank account number, internet banking credentials or One-Time Password.
Inheritance Scam	Victims receive a letter, call or email stating that they have been left a large fortune. However, to have the funds released, victims must first pay the administration fees and taxes.
Internet Love Scam	After befriending an attractive person (who is usually foreign) online, he or she tells a tale about falling into trouble or hard times. The scammer persists with the story to gain their victim's trust and adoration, then asks for money as proof of love. Once the money is transferred, the scammer disappears.
Investment Scam	Victims receive messages from people claiming to be stockbrokers or bank or financial company employees on social networking sites like Facebook, WeChat or Line. Responding to these messages leaves you vulnerable to an Investment Scam where fraudsters ask for personal details like NRIC and passport numbers, supposedly for an investment form. Scammers then ask victims to transfer money to banks in Hong Kong and China, pay administrative fees, security fees and taxes in order to receive the profits and returns. Victims also receive phone calls from people claiming to be from the Hong Kong Monetary Authority or Hong Kong Overseas Control Centre, asking for a deposit before profits can be released to them.

Table 1 continued from previous page

Scam Type	Description
Job Scam	These scams often involve online job ads seeking would-be male social escorts. Victims who respond are told they would be introduced to wealthy female clients, but only after they pay a registration fee. After payment, the scammers ask victims for other fees such as for insurance and membership before disappearing with the money. In another variation of the job scam, scammers place online job ads for assistant purchasers, stock takers or participants for a system trial on popular classified websites like Gumtree. Participants are asked to reveal personal details like their names, IC numbers, phone numbers, phone security codes and one-time passwords. Information like this allows scammers to access your mobile phone lines to purchase online credits. In another variant, scammers will offer jobs that require applicants to do the following in return for a small commission: (1) Processing fund transfers by receiving money into their personal bank accounts and then transferring the money out through online banking or money transfer services such as Western Union or MoneyGram; (2) Open bank accounts using their names for a business, or (3) Receive a donation into their personal bank accounts and assist to deposit the money into a crypto kiosk.
Kidnap Scam	Victims get a call from someone claiming that a loved one of the victim has been kidnapped. During the call, victims might hear screaming or crying in the background.
Line/Facebook Scam	Scammers hack into your Facebook or Line account and use your identity to ask contacts to buy iTunes or other gift cards for them. Conversely, you might be asked by a friend to buy the cards urgently.
Loan Scam	SMSes or WhatsApp messages are sent offering loans and loan services to random users. The scammers may claim to be staff from a licensed moneylender. Interested parties are instructed to transfer money as a deposit before the loan can be disbursed. After making the transfer, victims find that the scammers are no longer contactable. As part of these scams, the scammers may ask for personal information like NRIC and contact numbers, SingPass details and bank account numbers. When handed over, the information is used to harass or threaten victims for payment.

Table 1 continued from previous page

Scam Type	Description
Lottery Scam	Victims are approached while shopping and invited to participate in a simple scratch-and-win promotion. They win a prize but must follow the scammer to their main office to collect it. Once at the office, the winners/victims are told to pay an administration charge or tax on the prize, which does not exist. They might even be asked to pay more money to join a grand draw with prizes such as cars or holidays. The scammers offer to accompany "winners" to an ATM in Singapore to collect the money. Once they receive the money, scammers tell victims that the prize is delayed and that they should return another time. Victims receive a phone call or SMS notification that they have won a prize in a lucky draw. Often, the prize is a car or a condominium unit overseas. To claim the prize, victims must pay administrative fees or taxes. Or to convert the prize into cash, victims must make payment to a foreign bank account. In another variation of the lucky draw scam, victims receive a call to take part in a survey that qualifies them entry to a lucky draw. Victims subsequently win the draw, but must pay an administrative fee to claim the prize.
Money Mule Scam	Scammers, likely to be members of foreign syndicates, pose as lonely individuals seeking companionship and love online. They befriend victims on social media sites and after gaining their trust, ask the victims to open a new bank account or use an existing bank account to receive money. When the money is deposited into account, the victim is asked to pass or send the money to another person or company, usually based overseas. Alternatively, scammers post job advertisements on online job portals or social media platforms for the position of "agent". The "agents" will earn commission for receiving and transferring money for a "legitimate" company.
Online Purchase Scam	Victims of this scam are often tempted by what seems like a good deal for a gadget, amusement park or concert tickets sold online. They transfer payment to the "seller" who promises delivery of the item. In some cases, sellers demand further payment for duties or delivery charges after the first payment is made. Ultimately, the victim never receives the item.
Online Travel Vacation Scam	Scammers place online ads for a vacation at outrageously cheap prices. Upon leaving for the trip, victims find that the hotel accommodation and air travel they had paid for were never booked.
PayPal Email Scam	Scammers prey on sellers on online auction sites like Ebay and Carousell. They might agree to buy an item from the seller and make payment through PayPal. They then send the victim an email that looks like it had come from PayPal stating that the money has been sent. The seller is asked to provide shipment details or pay an admin fee before the payment can be released. Under the impression that it is a legitimate instruction from PayPal, the seller mails the item to the scammer.

Table 1 continued from previous page

Scam Type	Description
Phishing Scam	Victims receive a call informing them that they have won a lucky draw. To claim the prize, the victim must provide their passport details or other personal information. In another phishing scam, fake websites are created to look identical to the actual websites but with a slightly different web address. Should victims input their personal details and PIN numbers to these websites, their information and money are at risk.
Scam Using WeChat	Scammers use pop-up or online ads to sell game credits or Chinese Reminbi at attractive rates. These ads often appear in popular games. When a victim clicks on the pop-up ad, they will be instructed to add the seller on WeChat. Thereafter, the victim is asked to register an account on a website in order to receive the game credits. During registration, the victim is asked to provide their personal information and bank account details. When registration is complete, the victim is asked to make payment for the game credits via Alipay, iTunes or MyCard. Once payment is made, the scammer does not deliver the game credits and becomes uncontactable. Some victims are asked to make multiple payments for various fees or to authenticate or activate the bank account. In the sale of foreign currencies, scammers ask victims to transfer money to a Singapore bank account before they can receive the Chinese Renminbi in the victim's WeChat or Alipay account. Once the transfer is done, the scammer becomes uncontactable and blocks the victim on WeChat.
Software Update Scam	Victims receive a call from someone claiming that their computer is in need of a security or software upgrade. To get the upgrade, victims must give their software user account ID and password to the caller. Sometimes, victims are asked to type several commands onto their computer, after which their computer system falls under someone else's control. Alternatively, victims might be asked to purchase additional software online. When they do, the scammers take their credit card or bank account details for their own fraudulent use.

Table 1 continued from previous page

Scam Type	Description
Spoofed/Hacked Email Scam	A scammer impersonates a victim's supplier using a similar email address. The victim will be told to transfer money to a different bank account because the supplier's regular account has been suspended or is under audit. In another variant, scammers will hack into their victim's email account, that of the supplier's or business partner's. They will monitor the email correspondence between the two and at an opportune time, send an email to their victim to request for payment to be paid to another bank account. The spoofed email used by the scammer can closely mimic that of the original email address. In some cases, scammers may even use the same business logo, links to the company's website, or messaging format to trick their victims into believing that they have received a genuine request for payment. Victims will only come to realise that they have been scammed (often days later) when their actual suppliers call to inform them that they have not received their payment.
Wangiri Scam	This scam gets its name from the Japanese word Wangiri — 'wan' means 'one' and 'giri' means 'hang-up'. Victims receive a phone call from an overseas number, which rings just once. If they return the call, they will hear an advertisement for a subscription to a premium chat line or internet services. Victims are charged a premium for this call. In another variation of this scam, the caller, claiming to be an official, leaves a voice message informing the victim that there has been an emergency to which they must respond by calling them back. The latest version involves WhatsApp messages with contact attachments. Victims incur a hefty fee when they call that contact.

Appendix B: Acronyms in Our Text Corpus

Table 2: List of acronyms identified from text corpus and their unabbreviated forms

Acronym	Unabbreviated Form
ACRA	accounting and corporate regulatory authority
AKA	also known as
AMK	ang mo kio
ASAP	as soon as possible
ATM	automated teller machine
BF	boyfriend
CPF	central provident fund
CIMB	cimb bank
CB	circuit breaker
CMB	cmb bank
CID	criminal investigation department
DOB	date of birth
DBS	dbs bank
FB	facebook
GST	goods and services tax
HP	handphone
HQ	headquarters
HK	hong kong
HDB	housing development board
HSBC	hsbc bank
HR	human resource
ID	identity
NRIC	identity number
IC	identity number
I/C	identity number
ICA	immigration and checkpoints authority
IMO	in my opinion
IMDA	infocomm media development authority
IRAS	inland revenue authority of singapore
IG	instagram
ISD	internal security department
IBAN	international bank account number
ISP	internet service provider
KL	kuala lumpur
LMAO	laugh my ass off
LTD	limited
MBS	marina bay sands
MRT	mass rapid transit train
MOP	member of public
MOPS	members of public
MOF	ministry of finance
MOH	ministry of health
MINLAW	ministry of law
MOM	ministry of manpower
MIA	missing in action
MAS	monetary authority of singapore
NPC	neighbourhood police centre
OCBC	ocbc bank
OKC	okcupid
OTP	one time password
PRC	people republic of china
PC	personal computer
POSB	posb bank
PTE	private
PM	private message
SG	singapore
SPF	singapore police force
UK	united kingdom
USS	universal studios singapore
UOB	uob bank
WA	whatsapp
W/O	without
WP	work permit

Appendix C: Typographical Errors in Our Text Corpus

Table 3: List of typographical errors in our text corpus and their corrected forms

Misspelling	Corrected
a/c	account
abit	a bit
abt	about
acc	account
acct	account
admin	administrative
alot	a lot
amt	amount
ard	around
assent	accent
bcos	because
becareful	be careful
blk	block
bz	busy
carousel	carousell
cos	because
daugter	daughter
doc	document
docs	documents
dun	do not
dunno	do not know
fri	friday
frm	from
gf	girlfriend
gov	government
govt	government
harrass	harass
hv	have
I-banking	internet banking
impt	important
infomed	informed
informaing	informing
insta	instagram
instagam	instagram
instagramm	instagram
intl	international
juz	just
knowldge	knowledge
laywer	lawyer
linkein	linkedin
mandrain	mandarin
messanger	messenger
mon	monday
msg	message
msgs	messages
msia	malaysia
mths	months
nv	never
nvr	never
payapl	paypal
pls	please
plz	please
polive	police
rcvd	received
recieve	receive
recieved	received
scammmer	scammer
sg	singapore
shoppe	shopee
sing-tel	singtel
sintel	singtel
s'pore	singapore
suspecious	suspicious
suspicios	suspicious
suspicius	suspicious
suspicius	suspicious
suspicous	suspicious
suspision	suspicion
thru	through
thu	thursday
thur	thursday
thurs	thursday
tidner	tinder
tis	this
tix	tickets
tmr	tomorrow
tranfer	transfer
trans	transfer
transger	transfer
trf	transfer
tue	tuesday
tues	tuesday
wadsapp	whatsapp
watapps	whatsapp
watsapp	whatsapp
wed	wednesday
whastapp	whatsapp
whatapp	whatsapp
whatapps	whatsapp
whataspp	Whatsapp
whats app	whatsapp
whatsaap	whatsapp
whatsap	whatsapp
whatsapphe	whatsapp
whatsapps	whatsapp
whattapp	whatsapp
whattsapp	whatsapp
wikipeida	wikipedia
wk	week
ytd	yesterday

Appendix D: An Overview of a RNN Cell

The architecture of a RNN cell is shown in Figure 2.

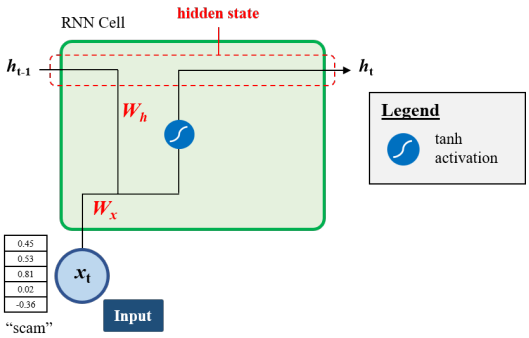


Figure 2: Architecture of a RNN cell

The hidden state of the previous time-step is h_{t-1} and that of the current time-step is h_t. The word embedding of the current time-step is given by x_t. Each RNN cell takes as input x_t and h_{t-1} and produces a hidden state, h_t. The underlying mathematical operation is as follows:

$$h_t = \tanh(W_h \cdot h_{t-1} + W_x \cdot x_t + b), \tag{1}$$

where W_h and W_x are weights associated with the hidden state and input respectively, and b is a bias term. With the hidden state of the current time-step, h_t, we can calculate the output, y_t, from this time-step using the following expression:

$$y_t = W_y \cdot h_t, \tag{2}$$

where W_y is the weight associated with the output.

Appendix E: An Overview of a LSTM Cell

Figure 3 shows the architecture of an individual LSTM cell. There are three types of gates, namely, forget gate, input gate and output gate. The explanation in this appendix is adapted from Olah [63].

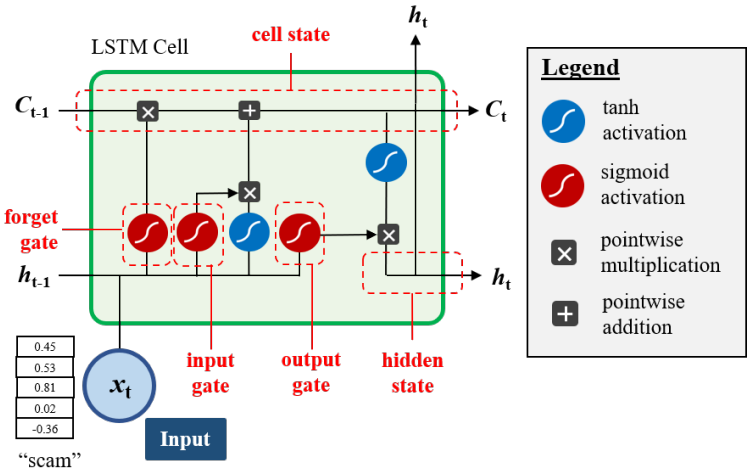


Figure 3: Architecture of a LSTM cell

Forget Gate

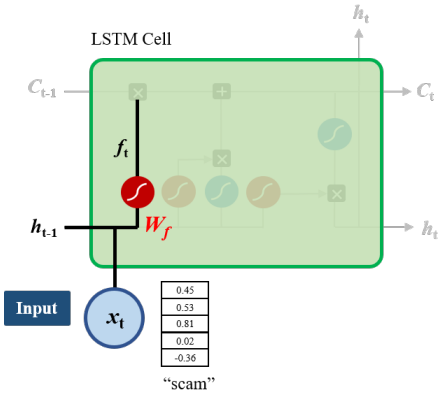


Figure 4: Forget gate

At the forget gate shown in Figure 4, the LSTM cell decides what information to discard from the cell state. First, the vector representing the hidden state of the previous time-step, h_{t-1}, is concatenated with the embedding of the input word at the current time-step, x_t, before being transformed with the sigmoid function. The mathematical operation is given by,

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \tag{3}$$

where W_f and b_f are the weights and bias associated with the forget gate respectively.

Input Gate

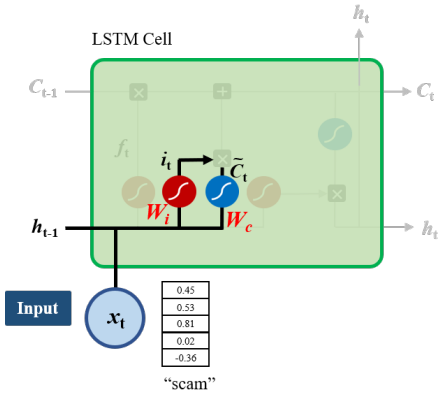


Figure 5: Input gate

At the input gate, the LSTM decides what new information to store in the cell state. This step is shown in Figure 5. There are two parts to this step. First, the input gate decides what information from the input to update. This uses another sigmoid function and is defined by,

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \tag{4}$$

where W_i and b_i are the weights and bias associated with with input gate respectively. Second, a tanh activation function produces a vector of new candidate values, $\tilde{C}_t$, which will be used to update the cell state. The mathematical operation of this is given by,

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C), \tag{5}$$

where W_C and b_C are the weights and bias before tanh activation function is applied. Following this, the cell state of the previous time-step, C_{t-1}, is updated using the following expression to derive the new cell state, C_t.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t. \tag{6}$$

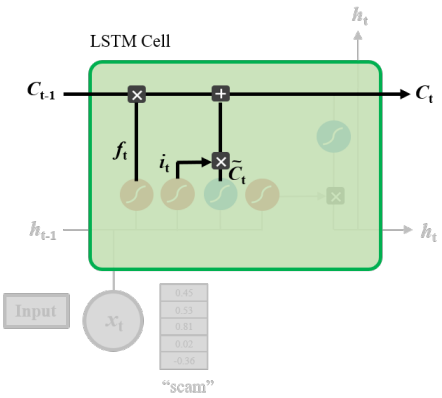


Figure 6: Input gate

Output Gate

Finally, at the output gate, the LSTM decides what information to output. The updated cell state is transformed by tanh activation function before being multiplied by the output of the output gate, o_t. The underlying mathematical operations are as follows:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \text{and} \tag{7}$$

$$h_t = o_t * \tanh(C_t). \tag{8}$$

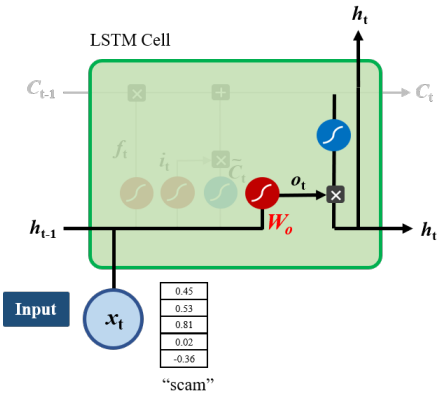


Figure 7: Output gate

Together, these three gates give LSTMs the ability to remove or add information from cell states, thereby allowing them to achieve good long-term dependencies across an input sequence.

Appendix F: Architectures of Deep Learning Models

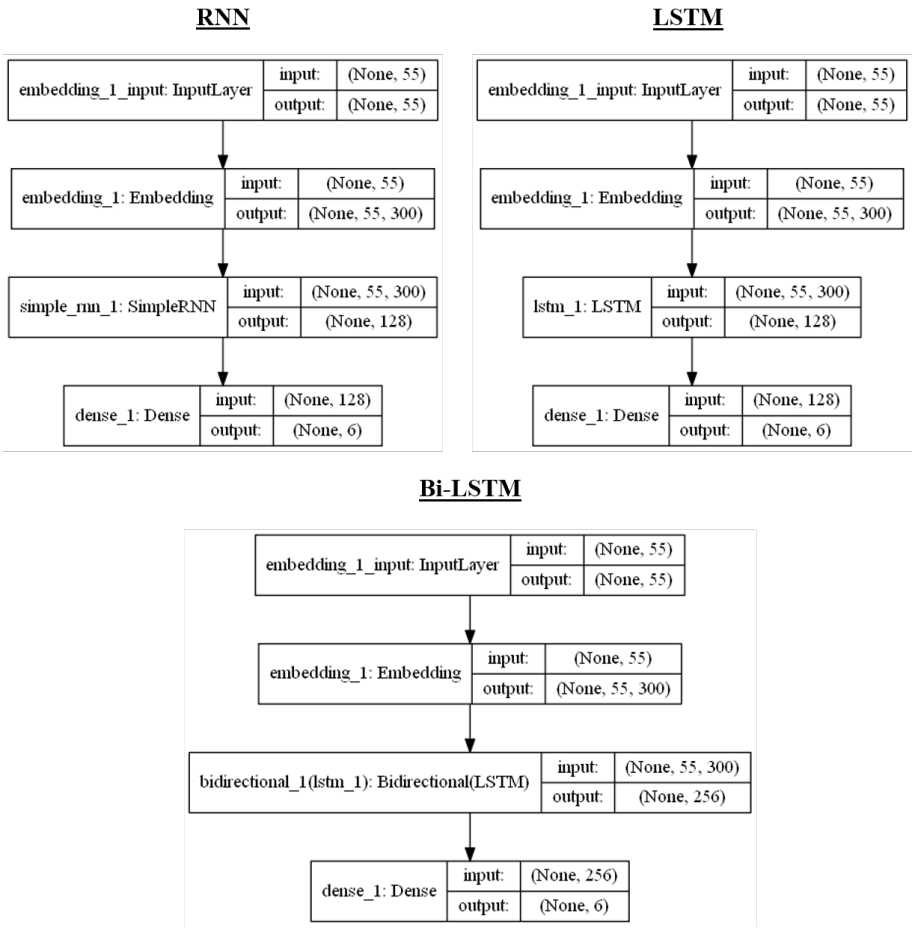


Figure 8: Architectures of RNN, LSTM, and Bi-LSTM used in model training

Appendix G: Summary of Parameters in Model Training

Table 4: Parameters for Experiments 1 and 3

	RNN		**LSTM**		**Bi-LSTM**	
	Without GloVe	**With GloVe**	**Without GloVe**	**With GloVe**	**Without GloVe**	**With GloVe**
Embedding dimension	300	300	300	300	300	300
Vocabulary size	9,092	9,092 (1,638 misses)	9,092	9,092 (1,638 misses)	9,092	9,092 (1,638 misses)
Dropout	0.1	0.1	0.1	0.1	0.1	0.1
Recurrent dropout	0.1	0.1	0.1	0.1	0.1	0.1
Input sequence length	55	55	55	55	55	55
Output dimensions of RNN/LSTM layer	128	128	128	128	256	256
Batch size	16	16	16	16	16	16
Optimiser	Adam	Adam	Adam	Adam	Adam	Adam
Learning rate	0.001	0.001	0.001	0.001	0.001	0.001
Total parameters	2,783,286	2,783,286	2,948,022	2,948,022	3,168,438	3,168,438
Trainable parameters	2,783,286	55,686	2,948,022	220,422	3,168,438	440,838

Table 5: Parameters for Experiment 2

	RNN		**LSTM**		**Bi-LSTM**	
	Without GloVe	**With GloVe**	**Without GloVe**	**With GloVe**	**Without GloVe**	**With GloVe**
Embedding dimension	300	300	300	300	300	300
Vocabulary size	6,992	6,992 (1,001 misses)	6,992	6,992 (1,001 misses)	6,992	6,992 (1,001 misses)

Table 5 continued from previous page

	RNN		LSTM		Bi-LSTM	
	Without GloVe	**With GloVe**	**Without GloVe**	**With GloVe**	**Without GloVe**	**With GloVe**
Dropout	0.1	0.1	0.1	0.1	0.1	0.1
Recurrent dropout	0.1	0.1	0.1	0.1	0.1	0.1
Input sequence length	66	66	66	66	66	66
Output dimensions of RNN/LSTM layer	128	128	128	128	256	256
Batch size	16	16	16	16	16	16
Optimiser	Adam	Adam	Adam	Adam	Adam	Adam
Learning rate	0.001	0.001	0.001	0.001	0.001	0.001
Total parameters	2,153,286	2,153,286	2,318,022	2,318,022	2,538,438	2,538,438
Trainable parameters	2,153,286	55,686	2,318,022	220,422	2,538,438	440,838

Appendix H: Precision and Recall Scores

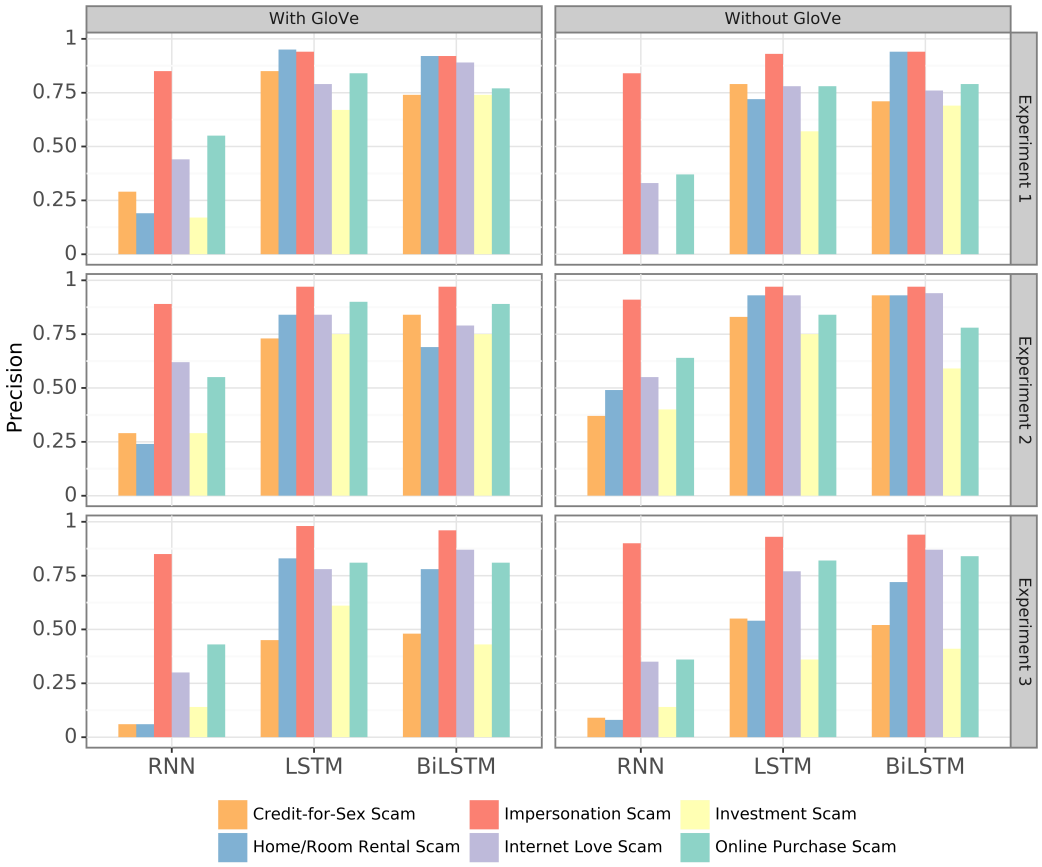


Figure 9: Results of experiments in terms of precision scores

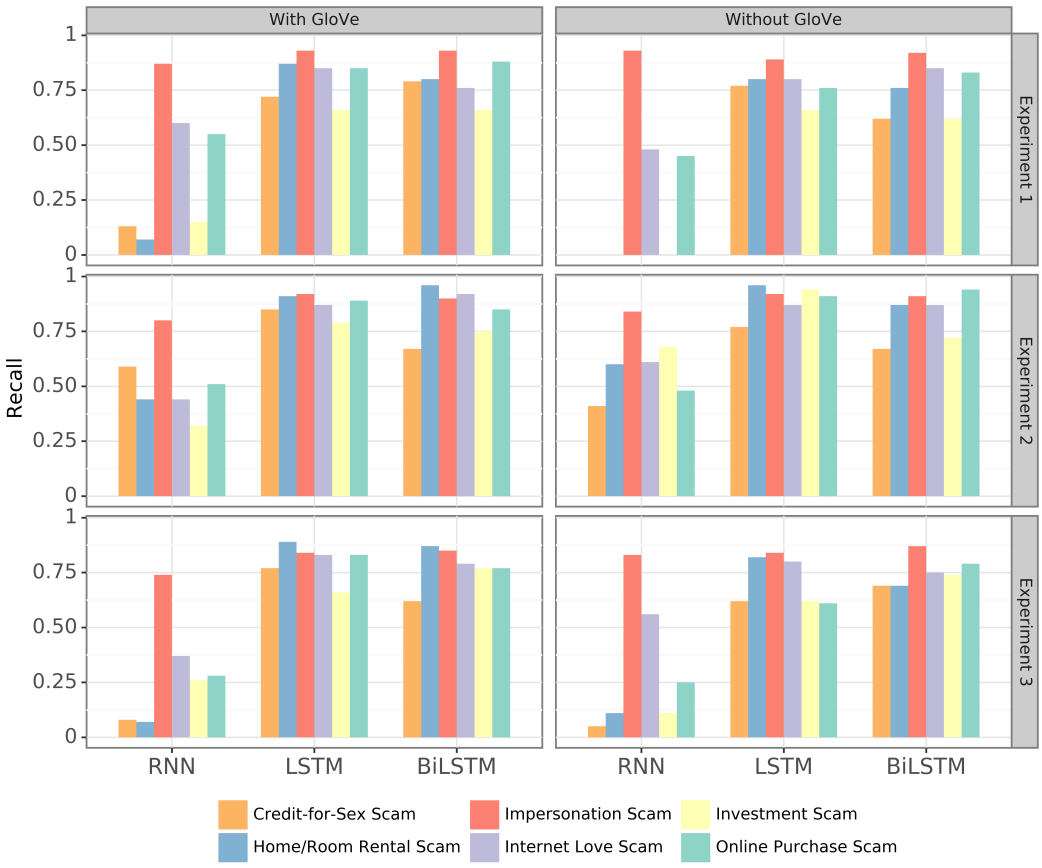


Figure 10: Results of experiments in terms of recall scores

Appendix I: Triplets of Candidate Documents

Table 6: Eight triplets of candidate documents from our text corpus

Triplet	Tag ID	Candidate Scam Reports (Pre-Processed Text)	Scam Type
1	157	automated phone call claiming it was from ministry of health with urgent information and to press 3 for more details. then a person came on the line speaking chinese, when i spoke english they hung up.	Impersonation Scam
	159	1 received a voice automated call from ministry of health asking me to follow their instructions as there is urgent information required by me. no money was lost.	Impersonation Scam
	2751	receiving calls recently from the number originating from switzerland, suspect it could be related to wangiri scam. usually is a missed call, but if attend i hear automated voice of a lady saying hello darling, which i believe is targeted for guys to attract and trigger to call back them to charge call money value. luckily read few articles about the scam and have blocked it.	Wangiri Scam
2	404	received a scam call today from. it was an indian female with a strong accent, claiming to be olivia smith from singtel calling to check on suspicious connections on my internet. she gave me a number to call back and said her employee identity was dcl. when i asked her to verify that she is a singtel officer, she hung up.	Impersonation Scam
	10	a lady claimed her name as olivia, calling from singtel tech department, employee number tlc. told me there is people stealing my internet, asked me to open my laptop terminal and execute netstat command, then explained to me what she made up how my ip has been misused by both domestic and foreign people. she was about to transfer me to another so called technical engineer before i hang up. she said she is authorized to resolve this today and asked me to call her back at. called my telco company and verified, this is totally a scam. do not trust the scammers.	Impersonation Scam

Table 6 continued from previous page

Triplet	Tag ID	Candidate Scam Reports (Pre-Processed Text)	Scam Type
	1841	since st august, i am getting numerous calls from this numbers. i have not pick up or called this number because its checked coming from lithuania country which never makes sense since i do not have any business or partners over there. i have blocked these numbers on my phone but apparently getting calls with with new extension numbers.advice please do not call back or receive call. its a wangiri scam dr.ptbn	Wangiri Scam
3	602	i got a call from dhl on dec. the number was at at am. when i picked up the call it was a machine recorded message saying that i have an unclaimed parcel and asked me to press number to speak to a staff. i immediately hung up after that.	Impersonation Scam
	711	received a call from dhl automated on nov approx. pm, saying i have unclaimed parcel. i hanged up immediately. do not press any number instructed by them, beware please hung up.	Impersonation Scam
	2944	booked for a rental car in australia and paid through the website. all seemed in order and info on the site stated that there were no extra charges.- upon arrival at the airport, there was no booking with the car company and they were not contactable. - went to the car rental company outlet and i had to pay full price again for the car.	Car Rental Scam
4	1161	i received a call from this with a name starhub company telling me that i won dollars. this guy was persuading me to get my money when i didnt even join any lucky draw and im not even using postpaid for my sim card. i quickly end the call. beware of scams.	Lottery Scam
	815	received calls on viber from starhub telecom, first call appeared to be an indian dialect i did not recognise. told them in english, they have a wrong number, say on the line - i hung up. they called back a minute later, a man with european accent, identified himself as from starhub. he asked if i knew starhub, then repeated my phone number back to me, and told me my number was entered in a lucky draw and i had just one. i said, no i did not, and that this is not real and i would report him to the police. then ask me where i was from i hung up again, blocked his number in viber and reported it here.	Lottery Scam
	2922	students and desperate job seekers beware of carousel job scam where they recruit you and ask you to pay you a one time deposit fee before hiring you. they are chinese scammers and if you dont do not understand chinese good for you. they will contact you on wechat. please anonymous, help us to eradicate these scammers be a hero and locate their ip address and then do a ddos attack that is all i want for christmas.	Job Scam

Table 6 continued from previous page

Triplet	Tag ID	Candidate Scam Reports (Pre-Processed Text)	Scam Type
5	3602	this guy claimed to be a south korean, united kingdom born raised, doesnt speak korean but neither do his english was as proficient. he is a very charming oppa like those from kdrama i knew from badoodating app. he only ask to chat via whatsapp united kingdom number. he narrated himself as a good guy then quickly claim that he is looking for marriage wants to involve you. next, asking for financial help in singapore. found another number auto-sync from whatsapp australia number.	Internet Love Scam
	3896	knew this guy from badoo, he claimed he is a german but was using an ireland country code phone to communicate. he said he is self-employed in boating business, travels a lot. recently he said he need to go japan then malaysia, 'while ' he was in malaysia, he started to tell me a lot about how he was made pay taxes and under-table money in the custom while he was trying to clear his cargo. he said he is tight in funds.....he is a scammer it is very recent, help get him.	Internet Love Scam
	1497	received an email from maybank informing me that i have million of inheritance fund waiting for me. but i need to pay fees and taxes to release my funds. i have been borrowing from my families and friends to pay for the taxes and fees. all amounted to about. i lied to my families and friends using my illness, children, husband, grandchildren as an excuse. i feel so ashamed of myself. this happened in november until this year.how can i share my full story.	Inheritance Scam
6	1755	this person texted me on. pm i kidnap your daughter, if you want your daughter safe. bank in now pos sav. dont report police, u report i will kill her.	Kidnap Scam
	1753	text message detail below i kidnap your daughter, if you want your daughter safe. bank in now pos sav dont report police, u report i will kill her.	Kidnap Scam
	2843	i was texted and had conversations for over a year with this woman every time we were to meet she came up with a different excuse. she claimed she loved me and sent me naughty pictures of herself claiming that she had never done that for anyone else. she also claimed to have an inheritance of gold which had to be released from a holding company. this is where my money went and for various other things.	Inheritance Scam

Table 6 continued from previous page

Triplet	Tag ID	Candidate Scam Reports (Pre-Processed Text)	Scam Type
7	322	i received a phone call today afternoon mar from an indian guy saying he is from microsoft support team. he is calling from. he first asked whether my name is correct and continued to say they received email notifications that my computer has issues, so microsoft called to help me resolve that issue. he wants me to go to my computer so he can guide me step-by-step on how to resolve that issue. that raised my suspicion so i questioned what exactly is the issue and how this issue came about. he replied saying it is because my computer do not have firewall thus it increases risk of having my password and sensitive information being leaked. thus he will need to guide me step by step on my computer and gain remote access to my computer to fix this. like hello i confirmed have firewall in place and giving you remote access is super dangerous, you can really then access all my password and sensitive information i rejected going to my computer and he wont give up by saying that he wants to schedule an appointment with me to resolved this issue. i said i was busy and hang up. afterwards i search official microsoft website on whether they will do this, turns out even on the official website, microsoft warns against having their staff calling customer for computer issues as their software are assured against these type of technical issues. everybody beware.	Impersonation Scam
	421	received a call from a indian man with very strong accent saying that he is calling from microsoft service. and that my computer has been compromise with more than devices trying to connect to my computer. he asked if i am near my computer. he ask to call me back later when i am beside my computer. i did not reply but in turn i asked him how he knew about it and he said he got the information from singtel. as the information he has given does not tally, when you received such phone calls, hang up.	Impersonation Scam
	758	a guy who claimed that he is from wyndham vacation and said that i am selected to get a special vacation treat on th october time, and mentioned that i need to answer pre-requisites questions asking me more of my personal details such as my marital status. later he also shared that i could just use sgd to choose special destination, bangkok, phuket or indonesia. when i told him i find it too good to be true and refuse to answer his questions that i find it is personal, he later said his time is precious and hence hung up. i later tried to call up to make sure that the number is valid, this time a lady picked up the call and told me another story about their company is promoting a great deal promotion but first i need to again answer the questions before i can proceed and before they sent an confirmation email to my mailbox. this time i chose to just hung up.if you received such cold calls, simply ignore and do not provide yoour info.	Online Travel Vacation Scam

Table 6 continued from previous page

Triplet	Tag ID	Candidate Scam Reports (Pre-Processed Text)	Scam Type
8	435	bought hand sanitizers on ninelif.sg but took awhile for me to realise it is a scam website after i tried calling the hotline number given on the website. the number was a residential number and indian lady on the other side of the phone mentioned that she received numerous calls everyday assing for delivery status. she said she is not in commercial business and was frustrated that she received so many nuisance calls from people enquiring on their ninelife.sg delivery status.i have asked payapl for a refund and hope that this can be claimed.beware of ninelif.sg site	Online Purchase Scam
	449	i purchased kids surgical mask on ninelif.sg on and payment has been made on the spot via paypal. order confirmation received. i just noticed that the merchant name on paypal record is headrus ventures pvt ltd. i followed up with the status on and received reply from sonali ninelif united_kingdom mentioned the product has been procured. she assured me will deliver it to me when it arrives singapore. on, nothing has been received. i emailed them again to follow up on the status. besides, i have also called to ninelif singapore for many times but it went to voice mail. i even called to ninelif united kingdom, but no one was picking up the calls. till date, i have not heard from them and nothing has been received. i am sure i had been scammed.editor is note masks and hand sanitizers are in high demand. please be wary when buying these items online. avoid advance payments or use platforms that offer escrow services to prevent getting scammed, and do research on the company or seller before transacting with them.	Online Purchase Scam

Table 6 continued from previous page

Triplet	Tag ID	Candidate Scam Reports (Pre-Processed Text)	Scam Type
	305	i am going to share on how the scam works. i have tried this out of curiosity to experience what the victims are going through. the credit for sex scammers usually do not have a local number tied to their advertisement, instead they will use a line-id or a wechat-id. the only way to connect with them is via these apps. they can advertise their services on locanto, or tinder or any other possible websites. the one i experienced was via locanto, and was directed to add this person on wechat. upon contacting the masseuse or the escort, they will pretend it is a legitimate proposition by asking you when you would like to meet them. once you have given them a date and a time, they will ask for your location. the location i was given was yishun avenue. upon arriving at the location, the girl will request for you to take a picture of your surroundings to indicate you have reached the meeting place. next, they will ask you for your contact number so that their friend can contact you with further instructions and direct you to their actual location.once you have given your contact number, their friend will call you via an unknown or private number, and ask you a series of questions. the questions asked will be whether or not this is your first time with this girl and if you have any friends or family who are in the police force. once they have verified that this is your first time and that you do not have any connections to the police, this is when they will proceed with the scam. he will then ask you to pay a deposit of or for the services which you requested via an axs machine, selecting the top up – alipay option with their e-mail given to you by the girl via the messaging app. this is the first red flag once you have done so, you will be requested by the girl to take a picture of the receipt and send via the app. next, the friend will then ask you to make a refundable deposit of as this is their company policy. they will justify it by saying that the girl is a student and is doing this illegally, the money will be used to bail them out in the event they are caught and deported back. do not transfer the money second red flag once you have rejected transferring the money, he will begin to threaten you by saying he have your contact number and will track you down and hurt your family. they will then negotiate with you by asking you to make a smaller amount for deposit. the girl will even call you on the messaging app to plead with you that it is for your safety. stay calm this is all a scam. stop topping up nor deposit any more money and report to the police.	Credit-for-Sex Scam

Bibliography

1. Singapore Police Force. *Police News Release: Annual Crime Brief 2016* (accessed on 3 August 2020). https://www.police.gov.sg/Media-Room/Statistics.
2. Singapore Police Force. *Police News Release: Annual Crime Brief 2018* (accessed on 3 August 2020). https://www.police.gov.sg/Media-Room/Statistics.
3. Singapore Police Force. *Police News Release: Annual Crime Brief 2017* (accessed on 3 August 2020). https://www.police.gov.sg/Media-Room/Statistics.
4. Singapore Police Force. *Police News Release: Mid-Year Crime Statistics* (accessed on 28 August 2020). https://www.police.gov.sg/Media-Room/Statistics.
5. Singapore Police Force. *Police News Release: Annual Crime Brief 2019* (accessed on 3 August 2020). https://www.police.gov.sg/Media-Room/Statistics.
6. Buchanan, T. & Whitty, M. T. The online dating romance scam: Causes and consequences of victimhood. *Psychology, Crime & Law* **20,** 261–283. doi:10.1080/1068316X.2013.772180 (2014).
7. Teh, C. *Woman loses $300k to 'Singtel customer service' caller helping her solve Wi-Fi connectivity problem.* The Straits Times. (accessed on 5 August 2020). https://www.straitstimes.com/singapore/courts-crime/woman-loses-300k-to-singtel-customer-service-caller-helping-her-solve-wi-fi (2019).
8. Ministry of Home Affairs. *Plan to Curb Increasing Trend of Scam Cases, Written Reply to Parliamentary Question by Mr K Shanmugam, Minister for Home Affairs and Minister for Law* (accessed on 5 August 2020). https://www.mha.gov.sg/newsroom/in-parliament/written-replies-to-parliamentary-questions (2019).
9. Teh, C. *Anti-Scam Centre recovers over $21.2m from cases such as love scams and bogus e-commerce dealings.* The Straits Times. (accessed on 5 August 2020). https://www.straitstimes.com/singapore/anti-scam-centre-recovers-over-212m-from-cases-including-love-scams-and-bogus-e-commerce (2020).
10. Teh, C. *New education campaign launched to address rising scam numbers* The Straits Times. (accessed on 28 August 2020). https://www.straitstimes.com/singapore/new-education-campaign-launched-to-address-rising-scam-numbers (2020).

11. Brown, K. & Carter, E. Scams: The power of persuasive language. doi:10.13140/RG.2.2.31456.51209 (2020).

12. Norris, G., Brookes, A. & Dowell, D. The Psychology of Internet Fraud Victimisation: a Systematic Review. *Journal of Police and Criminal Psychology* **34,** 231–245. doi:10.1007/s11896-019-09334-5 (2019).

13. Van Rossum, G. & Drake, F. L. *Python 3 Reference Manual* ISBN: 1-4414-1269-7 (CreateSpace, Scotts Valley, CA, 2009).

14. Renelle, T. *Natural Language Processing.* OCDevel Podcast: Machine Learning Guide, Episode 18. (accessed on 31 July 2020). http://ocdevel.com/mlg/18.

15. Chau, M., Xu, J. J. & Chen, H. *Extracting Meaningful Entities from Police Narrative Reports* in *Proceedings of the 2002 Annual National Conference on Digital Government Research* (2002).

16. Al-Zaidy, R., Fung, B. C., Youssef, A. M. & Fortin, F. Mining criminal networks from unstructured text documents. *Digital Investigation* **8,** 147–160. doi:10.1016/j.diin.2011.12.001 (2012).

17. Elyezjy, N. T. & Elhaless, A. M. Investigating Crimes using Text Mining & Network Analysis. *International Journal of Computer Applications* **126.** doi:10.5120/ijca2015906134 (2015).

18. Schraagen, M., Testerink, B., Odekerken, D. & Bex, F. *Argumentation-driven information extraction for online crime reports* in *CIKM Workshops* (2018).

19. Bex, F., Peters, J. & Testerink, B. *A.I. for Online Criminal Complaints: From Natural Dialogues to Structured Scenarios* in *Artificial Intelligence for Justice Workshop (ECAI 2016)* (2016).

20. Testerink, B. & Bex, F. *Demo: Natural Language Processing for Online Fraud Scenario Extraction* in *Artificial Intelligence for Justice Workshop (ECAI 2016)* (2016).

21. Yiu, T. *Understanding NLP and Topic Modeling Part 1* (accessed on 2 August 2020). https://www.kdnuggets.com/understanding-nlp-and-topic-modeling-part-1.html/ (2019).

22. Salgado, R. *Topic Modeling Articles with NMF* (accessed on 2 August 2020). https://towardsdatascience.com/topic-modeling-articles-with-nmf-8c6b2a227a45 (2020).

23. Kuang, D., Brantingham, P. J. & Bertozzi, A. L. Crime Topic Modeling. *Crime Science* **6,** 12. doi:10.1186/s40163-017-0074-0 (2017).

24. Birks, D., Coleman, A. & Jackson, D. Unsupervised Identification of Crime Problems from Police Free-text Data. doi:10.31235/osf.io/8w73n (2020).

25. Blei, D. M., Ng, A. Y. & Jordan, M. I. Latent Dirichlet Allocation. *Journal of Machine Learning Research* **3,** 993–1022. https://www.jmlr.org/papers/v3/blei03a (2003).

26. Abu-Nimeh, S., Nappa, D., Wang, X. & Nair, S. *A Comparison of Machine Learning Techniques for Phishing Detection* in *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit* (2007).

27. Mbaziira, A. & Jones, J. *A Text-based Deception Detection Model for Cybercrime* in *International Conference on Technology and Management* (2016).

28. Mbaziira, A. V., Abozinadah, E. & Jones Jr, J. H. Evaluating Classifiers in Detecting 419 Scams in Bilingual Cybercriminal Communities. *arXiv preprint arXiv:1508.04123* (2015).

29. Mohammad, R. M. A., McCluskey, T. L. & Thabtah, F. *Predicting Phishing Websites using Neural Network trained with Back-Propagation* in *World Congress in Computer Science, Computer Engineering, and Applied Computing* (2013).

30. Bahnsen, A. C., Bohorquez, E. C., Villegas, S., Vargas, J. & González, F. A. *Classifying phishing URLs using recurrent neural networks* in *2017 APWG Symposium on Electronic Crime Research (eCrime)* (2017), 1–8. doi:10.1109/ECRIME.2017.7945048.

31. Chatterjee, M. & Namin, A.-S. *Detecting Phishing Websites through Deep Reinforcement Learning* in *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)* **2** (2019), 227–232. doi:10.1109/COMPSAC.2019.10211.

32. Feng, J., Zhang, Y. & Qiao, Y. A Detection Method for Phishing Web Page Using DOM-Based Doc2Vec Model. *CIT. Journal of Computing and Information Technology* **28,** 19–31 (2020).

33. Wu, J. *et al.* Who Are the Phishers? Phishing Scam Detection on Ethereum via Network Embedding. *arXiv:1911.09259* (2019).

34. LOI English. *A Fraud and A Scam: Do You Know The Difference?* (accessed on 29 July 2020). https://www.skypeenglishclasses.com/fraud-scam-know-difference-3/ (2017).

35. Pouryousefi, S. & Frooman, J. The Consumer Scam: An Agency-Theoretic Approach. *Journal of Business Ethics* **154,** 1–12. doi:10.1007/s10551-017-3466-x (2019).

36. Attorney-General's Chambers. *Penal Code - Cheating* (accessed on 3 August 2020). https://sso.agc.gov.sg/Act/PC1871?ProvIds=P4XVII-P4_415- (2020).

37. Cohen, L. E. & Felson, M. Social Change and Crime Rate Trends: A Routine Activity Approach. *American Sociological Review* **44,** 588–608 (1979).

38. Wilsem, J. v. Worlds tied together? Online and non-domestic routine activities and their impact on digital and traditional threat victimization: *European Journal of Criminology* **8,** 115–127. doi:10.1177/1477370810393156 (2011).

39. Stanford Center on Longevity. *Low Self-Control, Routine Activities, and Fraud Victimization* (accessed on 3 August 2020). http://longevity.stanford.edu/2011/06/17/low-self-control-routine-activities-and-fraud-victimization/ (2011).

40. Pattinson, M., Jerram, C., Parsons, K., Mccormac, A. & Butavicius, M. *Managing Phishing Emails: A Scenario-Based Experiment* in *Human Aspects of Information Security & Assurance* (2011), 74–85.

41. Fischer, P., Lea, S. E. G. & Evans, K. M. Why do individuals respond to fraudulent scam communications and lose money? The psychological determinants of scam compliance. *Journal of Applied Social Psychology* **43,** 2060–2072. doi:10.1111/jasp.12158 (2013).

42. Wright, R. & Marett, K. The Influence of Experiential and Dispositional Factors in Phishing: An Empirical Investigation of the Deceived. *Journal of Management Information Systems* **27,** 273–303. doi:10.2753/MIS0742-1222270111 (2010).

43. Cornish, D. B. & Clarke, R. V. *The Reasoning Criminal: Rational Choice Perspectives on Offending* ISBN: 978-1-4128-5229-6 (Transaction Publishers, 1986).

44. Yee, Z. W., Yeh, V., Ong, S. & Han, Y. Stealing More Than Just Your Heart: A Preliminary Study of Online Love Scams. *Home Team Journal - By Practioners, For Practioners* (2019).

45. Friedman, D. A. Imposter Scams. *Social Science Research Network Electronic Journal.* doi:10.2139/ssrn.3536026 (2020).

46. Wang, J., Herath, T., Chen, R., Vishwanath, A. & Rao, H. R. Research Article Phishing Susceptibility: An Investigation Into the Processing of a Targeted Spear Phishing Email. *IEEE Transactions on Professional Communication* **55,** 345–362. doi:10.1109/TPC.2012.2208392 (2012).

47. Luo, X., Zhang, W., Burd, S. & Seazzu, A. Investigating phishing victimization with the Heuristic Systematic Model: A theoretical framework and an exploration. *Computers & Security* **38,** 28–38. doi:10.1016/j.cose.2012.12.003 (2013).

48. Graham, R. & Triplett, R. Capable Guardians in the Digital Environment: The Role of Digital Literacy in Reducing Phishing Victimization. *Deviant Behavior* **38,** 1371–1382. doi:10.1080/01639625.2016.1254980 (2017).

49. Smith, R. G. & Jorna, P. Fraud in the 'outback': Capable guardianship in preventing financial crime in regional and remote communities. *Trends and Issues in Crime and Criminal Justice* **413** (2011).

50. Cornish, D. *The Procedural Analysis of Offending and Its Relevance for Situational Prevention* in *Crime Prevention Studies* (Criminal Justice Press, 1994), 151–196.

51. Poh, A. National Crime Prevention Council, personal communication, 17 June 2020.

52. Leonard, R. *Beautiful soup documentation* (2007).

53. Software Freedom Conservancy. *Selenium Webdriver documentation* (2013).

54. Norvig, P. *pyspellchecker* (2018).

55. Bird, S. & Loper, E. *Natural Language Toolkit* (2001).

56. Bitext. *Lemmatization vs Stemming* (accessed on 6 August 2020). https://blog.bitext.com/lemmatization-vs-stemming (2016).

57. Bengfort, B., Bilbro, R. & Ojeda, T. *Applied Text Analytics with Python: Enabling Language Aware Data Products With Machine Learning* ISBN: 978-1-4919-6304-3 (O'Reilly, 2018).

58. Honnibal, M. & Montani, I. *spaCy: Industrial-strength Natural Language Processing in Python* (2017).

59. Ministry of Home Affairs. *Scam Cases Reported Since the COVID-19 Started, Written Reply to Parliamentary Question by Mr K Shanmugam, Minister for Home Affairs and Minister for Law* (accessed on 5 August 2020). https://www.mha.gov.sg/newsroom/in-parliament/written-replies-to-parliamentary-questions (2019).

60. Aghabozorgi, S. *Introduction to Machine Learning.* Coursera: Machine Learning with Python. (accessed on 25 July 2020).

61. Provost, F. & Fawcett, T. *Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking* ISBN: 978-1-4493-6132-7 (O'Reilly Media, 2013).

62. Shickel, B., Tighe, P., Bihorac, A. & Rashidi, P. Deep EHR: A Survey of Recent Advances in Deep Learning Techniques for Electronic Health Record (EHR) Analysis. *IEEE Journal of Biomedical and Health Informatics* **22,** 1589–1604. doi:10.1109/JBHI.2017.2767063 (2018).

63. Olah, C. *Understanding LSTM Networks* (accessed on 9 August 2020). https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (2015).

64. Johnson, J. M. & Khoshgoftaar, T. M. Survey on deep learning with class imbalance. *Journal of Big Data* **6,** 27. doi:10.1186/s40537-019-0192-5. https://doi.org/10.1186/s40537-019-0192-5 (2019).

65. Wang, S. & Yao, X. Multiclass Imbalance Problems: Analysis and Potential Solutions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **42,** 1119–1130. doi:10.1109/TSMCB.2012.2187280 (2012).

66. Wei, J. & Zou, K. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. *arXiv preprint arXiv:1901.11196* (2019).

67. Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* **16,** 321–357. doi:10.1613/jair.953 (2002).

68. Kunert, R. *Synthetic Minority Over-sampling TEchnique line by line.* (accessed on 9 August 2020). https://rikunert.com/SMOTE_explained (2017).

69. Subramanian, V. *Deep Learning with PyTorch: A practical approach to building neural network models using PyTorch* ISBN: 978-1-78862-433-6 (Packt Publishing Ltd, 2018).

70. Bengio, Y., Ducharme, R. & Vincent, P. in *A Neural Probabilistic Language Model: Advances in Neural Information Processing Systems 13* 932–938 (MIT Press, 2001). ISBN: 978-0-262-12241-2.

71. Mikolov, T., Chen, K., Corrado, G. & Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*. doi:https://arxiv.org/abs/1301.3781v3 (2013).

72. Linguistic Data Consortium. *English Gigaword Fifth Edition* (accessed on 9 Aug 2020). https://catalog.ldc.upenn.edu/LDC2011T07 (2011).

73. Pennington, J., Socher, R. & Manning, C. *Glove: Global Vectors for Word Representation* in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Association for Computational Linguistics, 2014), 1532–1543. doi:10.3115/v1/D14-1162.

74. Chollet, F. *Keras* (2015).

75. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **12,** 2825–2830 (2011).

76. Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980*. http://arxiv.org/abs/1412.6980 (2014).

77. Bishop, C. *Pattern Recognition & Machine Learning* ISBN: 978-0-387-31073-2 (Springer, New York, 2006).

78. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* **15,** 1929–1958 (2014).

79. Lemaitre, G., Nogueira, F. & Aridas, C. K. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research* **18,** 1–5. http://jmlr.org/papers/v18/16-365 (2017).

80. Graves, A., Jaitly, N. & Mohamed, A.-r. *Hybrid speech recognition with Deep Bidirectional LSTM* in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding* (2013), 273–278. doi:10.1109/ASRU.2013.6707742.

81. Wallace, B. C., Small, K., Brodley, C. E. & Trikalinos, T. A. *Class Imbalance, Redux* in *2011 IEEE 11th International Conference on Data Mining* (2011), 754–763. doi:10.1109/ICDM.2011.33.

82. Blagus, R. & Lusa, L. SMOTE for high-dimensional class-imbalanced data. *BMC Bioinformatics* **14,** 106. doi:10.1186/1471-2105-14-106 (2013).

83. Raj, B. S. *Understanding BERT: Is it a Game Changer in NLP?* (accessed on 26 August 2020). https://towardsdatascience.com/understanding-bert-is-it-a-game-changer-in-nlp-7cca943cf3ad (2019).

84. Vaswani, A. *et al.* in *Advances in Neural Information Processing Systems 30* 5998–6008 (Curran Associates, Inc., 2017). http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

85. Wittgenstein, L. *Philosophical Investigations* ISBN: 0-631-11900-0 (Basil Blackwell, Oxford, United Kingdom, 1953).

86. Jurafsky, D. & Martin, J. H. in *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition* (2019). ISBN: 978-0-13-187321-6.

87. Osgood, C. E., Suci, G. J. & Tannenbaum, P. H. *The measurement of meaning* (University of Illinois Press, Oxford, England, 1957).

88. Le, Q. V. & Mikolov, T. *Distributed Representations of Sentences and Documents* in *International conference on machine learning* (2014), 1188–1196. http://arxiv.org/abs/1405.4053.

89. Řehůřek, R. & Sojka, P. *Software Framework for Topic Modelling with Large Corpora* in *LREC 2010 workshop New Challenges for NLP Frameworks* (ELRA, Valletta, Malta, 2010), 45–50. ISBN: 978-2-9517408-6-0.

90. Mohr, G. *Avoiding over-fitting in Doc2Vec,* Gensim developer, personal communication, 21 August 2020. https://groups.google.com/g/gensim/c/JtUhgUjx4YI/m/3tvXgnSgBgAJ.

91. Řehůřek, R. *Doc2Vec Model* (accessed on 14 July 2020). https://radimrehurek.com/gensim/auto_examples/tutorials/run_doc2vec_lee.html (2019).

92. Ng, H. *Supreme Court to lodge police report after public feedback on scammers impersonating court officers.* The Straits Times. (accessed on 23 August 2020). https://www.straitstimes.com/singapore/supreme-court-to-lodge-police-report-after-public-feedback-on-scammers-impersonating-court (2018).

93. Eber, A. *Supreme Court warns of rise in scam calls, phishing email impersonating court officers* TODAY Online. (accessed on 23 August 2020). https://www.todayonline.com/singapore/supreme-court-warns-rise-scam-calls-phishing-emails-impersonating-court-officers (2020).

94. Plotly Technologies Inc. *Collaborative data science* (2015).