NOTTINGHAM TRENT UNIVERSITY

SCHOOL OF SCIENCE AND TECHNOLOGY

Email Spam Filter

by

Tomiwa Oladejo

in

2023

Project report in part fulfilment

of the requirements for the degree of

Bachelor of Science with Honours

In

Cyber Security

I hereby declare that I am the sole author of this report. I authorize the Nottingham Trent University to lend this report to other institutions or individuals for the purpose of scholarly research.

I also authorize the Nottingham Trent University to reproduce this report by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Signature

Tomiwa Oladejo

Abstract

Email spam has become a significant problem for both individuals and organizations, resulting in increased cybersecurity risks, loss of productivity, and cluttered inboxes. This study aims to develop an effective email spam filter using deep learning techniques to classify emails as either spam or ham.

The research uses a dataset containing labelled spam and ham emails, pre-processing techniques to clean and tokenize the text data, and a neural network model for classification. The pre-processing steps involve removing stop-words, punctuations, and subject lines to retain only relevant information for classification. The data is then tokenized and transformed into sequences of equal length. The deep learning model is compromised of an embedding layer, a flatten layer, and a dense output layer. The model is trained and evaluated using a train-test split approach and its performance is assessed based on accuracy and AUC metrics.

The results show promising performance in detecting and filtering spam emails, with potential applications in small businesses and individual email clients. Future work could involve incorporating more advanced deep learning architectures, exploring additional feature engineering techniques, and integrating the model into email systems for real-time spam filtering.

ACKNOWLEDGEMENTS

I would like to acknowledge and thank my supervisor Richard Otuka for guiding me throughout my project.

Table of Contents

Note: To insert TOC here follow the instruction below. You may also right click on the above text and use "Update Field" to update the TOC.

Insert > Reference > Index and Tables … > Table of Contents

List of Figures

Note: To insert List of Figures here follow the instruction below. You may also right click on the above text and use “Update Field” to update the list of figures.

Insert > Reference > Index and Tables … > Table of Figures > Caption Label => Figure

List of Tables

Note: To insert List of Tables here follow the instruction below. You may also right click on the above text and use "Update Field" to update the list of tables.

Insert > Reference > Index and Tables … > Table of Figures > Caption Label => Table

CHAPTER 1

Introduction

1.1 Introduction

Email spam tends to harbour certain characteristics. Numerous companies frequently send promotional emails to a large number of recipients as an attempt to promote their services and products. These emails may also contain various forms of phishing scams, malware and spoofing threats. The user's personal information is often exposed through these methods. The device being opened up to dangerous software can lead to the exposure of the user's personal information which is inclusive of passwords, addresses and bank details.

The aim of this project is to create an email spam filter capable of analysing and detecting unsolicited emails which may cause harm. In order to achieve this goal, research will be carried out throughout the course of the project to discern the necessary functionalities required to produce a practical solution.

Objectives:

- Investigate existing solutions and propose gaps regarding state of art
- Develop an existing algorithm that can detect and filter spam emails

In order to outline the requirements for the project, research will be carried out to build an understanding of the foundation necessary to construct a functional email spam filter. Administration of the research will also encompass a look into beneficial functionalities which may be implemented into the program throughout the process. Additionally, an investigation will take place regarding the existing solutions to the task. The analysis of these solutions will be inclusive

of various methods containing algorithms which may be disparate. This approach will assist the generation of a broader perspective in relation to the scope of the project. Inspiration may also be extracted from the investigation, allowing the integration of valuable functionalities, leading to a well-suited and structured solution.

Development of the program will be carried out using the accumulated data gathered throughout the research, with python as the main programming language. The benefits and performances of the different techniques will be evaluated to build an adequate solution. After the completion of the program, a series of tests will be executed to create an evaluation of the solution.

The subsequent chapters consist of the following:

Context - The context will contain a literature review, detailing the State-of-the-Art in the field. Additional research will also be carried out to identify the limitations residing within existing solutions. This may include factors such as the methods and algorithms used.

New Ideas - Within new ideas, a new proposition to reduce previous existing limitations will be introduced.

Implementation - The implementation will follow the process of the creation of the program. All the development and implementation of ideas will be documented, along with any adversities that surfaced throughout the task.

Results - A chapter displaying the significance of the results in conjunction with the new ideas.

Conclusion - To conclude the report, a summary of the project achievements and the how these may lead to future work will be reported. A synoptic assessment

will also be included, showing reflection on the skills developed for the duration of the project.

CHAPTER 2

CONTEXT

2.1 Introduction

With billions of emails being sent and received daily, emails are an increasingly important form of communication. However, as the use of emails grows, the influx of spam emails increases equally. These spam emails often contain harmful content such as malware within attachments or links to malicious websites. Using email spam filters, users are able to manage their email inbox in order to protect themselves from spam. The aim of this literature review is to provide a comprehensive overview of email spam filters. This is inclusive of the techniques, types, trends, challenges, performance evaluation and recent developments.

2.2 Literature Review

2.2.1 Introduction

Email spam filters are software programs capable of automatically scanning incoming emails and discerning whether or not the email is spam. The purpose of these filters is to reduce the quantity of undesired emails within the user's inbox. There are various techniques employed in email spam filters available which are studied below.

2.2.2 Pre-processing

The role of pre-processing is to prepare the raw email data for feature extraction and classification, achieved by the removal of irrelevant information. Various techniques are applied for the removal of irrelevant information.

One pre-processing technique commonly used in email spam filters is tokenization. Tokenization involves breaking down the email content into small components known as tokens. These tokens can be analysed for features such as word frequency and word co-occurrence, which are used to identify spam emails (Hassan and Mtetwa, 2018). Stop-word removal is another technique employed in spam filters. Common words such as “and”, “a” and “the” are known as stop-words. These words hold little significance in the determining the meaning of an email. An additional technique used for pre-processing is stemming. This involves the reduction of words to their root form (Ismail et al., 2022). An example would be the conversion of jogging to “jog”. Finally, feature selection is used to select a subset of relevant features from the dataset. The selected features are then used in the classification. Implementing these features, can improve the efficiency and accuracy of classification.

(Ruskanda, 2019) investigates the impact of various pre-processing techniques. These are inclusive of stop-word removal, tokenization, stemming and feature selection. Throughout the study, the efficiency of these methods are evaluated using the machine learning methods Naïve Bayes and SVM. Tokenization, feature selection and stop-word removal were found to positively improve the classification accuracy. However, stemming was found to have a minor impact on the overall performance.

(Ramón Méndez Reboredo, 2005) focused on the effects of stemming, tokenization and stop-word removal in conjunction with the naive bayes classifier. Similar to (Ruskanda, 2019), the combination of pre-processing methods led to a significant increase in the classification's accuracy. However, (Ruskanda, 2019) study claimed the stemming had a low impact.

Both studies found the implementation of pre-processing techniques can cause an increase in the accuracy of email spam detection. Although the authors agree that stop-word removal and tokenization have a positive impact, they concur on the effect of stemming. This discrepancy may be due to the use of different classifiers, datasets and languages.

A major advantage of pre-processing is the improved quality of data caused by the removal of inconsistencies and errors. This leads to improved accuracy of data analysis and standardization of data. The size of data can also be reduced, making it more efficient and manageable for the model to use. On the other hand, information can potentially be lost during the process. The quality of the raw data can also affect the quality of the pre-processing, meaning if the data quality is poor, the pre-processing will be too. This is why it is important for the techniques to be selected based on the requirements.

2.2.3 Rule-based filters

Rule-based filters: This filter uses pre-defined conditions or rules to separate and filter spam from legitimate emails, blocking unwanted messages. The complexity of the rules can vary and focus on inspecting specified characteristics of the

email. These characteristics encompass checking specific types of attachments, domain or email addresses and in certain words or phrases residing in the email body or subject (Reddy Guda, 2022). Although rule-based filters are simple and effective, it may not be adept at catching new or previously unknown types of spam.

Blacklist filtering: Blacklist filtering is a technique identify and maintain a list comprised of known sources of spam (domains, email addresses, IP addresses) and then block emails from these sources (Sharma et al. (2018). As an email is received, the message is analysed against the contents of the blacklist. If there are any matches between the message and the blacklist, it is labelled as spam and either moved to the spam folder or blocked.

Whitelist filtering accommodates a list of trusted sources that have been judged as safe (Sharma et al. (2018). If the message originates from these sources, it is allowed to bypass the spam filter and be delivered into the recipient's inbox. The database consists of sources such as IP addresses, email addresses and domains.

In (Saidani, 2020) study, a semantic-based classification leveraging rule-based filters is proposed. The method incorporates semantic information using the semantic relations between words (e.g. homophones, homographs and homonyms). A term frequency-inverse document frequency (TF-IDF) was also used. The machine learning model SVM was used with Naïve Bayes and k-Nearest Neighbours. A superior performance was produced by the classification approach in comparison to traditional feature extraction techniques.

(Shrivastava, 2014)focused on the application of adaptive genetic algorithms (AGA) for email spam filtering. The algorithmic approach employed by the

author, searches for the most relevant features that improve classification accuracy. The study's main focus was feature selection. However, the author states the use of rule-based filters is an essential step in pre-processing.

The applicable use of rule-based filters in conjunction with other advanced feature extraction and selection methods can enhance the performance of email spam filters is highlighted by both studies.

An advantage of rule-based filtering is the effective simplicity, making it easy to use. As the rules are pre-defined, computational resources are not required for the examination of messages. The rules can also be customised to meet individual user requirements. However, the filter may be liable to false negatives and false positives due to its reliance on pre-defined rules. An example of this scenario would be if a rule was set to block messages containing the word "buy", legitimate emails with the word could also be blocked. More sophisticated forms of spam which include emails that utilize social engineering tactics or phishing attacks to deceive users, are also factors the filter may not be effective against.

2.2.4 Machine Learning

Machine learning-based filters, also known as AI-based filters, utilize advanced algorithms to learn and adapt to new forms of spam over time, enabling the filter to automatically detect spam emails. Large volumes of email data are examined and categorized using machine learning techniques comprised of Deep Learning, Naïve Bayes and Support Vector Machines (SVM). The information is then used to discern and block spam messages.

The Naïve Bayes algorithm uses the Bayes' theorem to determine whether or not an email is spam (V.Christina, 2010). The theorem is a probabilistic

mathematical formula which calculates the probability of a hypothesis based on formerly analysed data.

It functions by using a dataset of emails, inclusive of non- spam and spam, to analyse the statistical patterns within the messages. The algorithm uses features such as the subject line, sender and the content to learn the relationships between the data. After the information has been gathered, it is employed to classify and separate spam and non-spam (Kumar, Sonowal and Nishant, 2020). The conditional probability of each feature given a class label is calculated before they are multiplied together, resulting in the probability of the email being associated with either class. These probabilities are compared to deduce the favoured classification. The calculations involved in determining the probability of the message being spam or legitimate are simplified as the algorithm assumes that particular email features are not connected to each other (Kumar, Sonowal and Nishant, 2020). Although this is not always accurate in real scenarios, it is beneficial. Naïve Bayes is a popular algorithm for email spam filtering as it can be trained quickly and can achieve high accuracy rates.

An advantageous point of the algorithm is its relative simplicity combined with the system’s ability to adapt at a rapid pace. Even if the provided training data is limited, it can still be effective. Moreover, the Naïve Bayes algorithm is convenient for handling email data as it is capable of operating with a wide spectrum of features. On the other hand, when it comes to real-world scenarios, the algorithms presumption of particular email features being independent between each other can lead to cases of inaccurate classification. Additionally, the algorithm may become over adapted to the dataset whilst training. This can cause a poor performance when used on new, unseen data.

Support Vector Machines (SVM) is another machine learning algorithm which creates a hyperplane used to separate two classes of data points (non-spam and spam messages) within a high-dimensional feature space (Awad, 2011). The algorithm accomplishes this by first learning form a dataset of emails used to train the machine. Each email within the dataset has a set classification. Using this data, SVM identifies the best hyperplane between the two classes. This method allows the margin amidst spam and non-spam to be maximized, resulting in a reduced probability of new emails being misclassified with a better separation between the classes.

The SVMs ability to transform data into a higher-dimensional feature space grants an increase in the accurate classification of emails (Cinelli et al., 2017)). Furthermore, the algorithm has the ability to handle non-linear decision boundaries, meaning the complex relationships connecting email features and its classification can be encapsulated. Additionally, SVMs can be used with different kernel functions to map the data to a higher-dimensional (spaceMa, 2020). A linear hyperplane can be separate the classes more effectively within this space.

A substantial limitation of Support Vector Machines (SMVs) is how expensive an investment it may be. Due to the requirement of a significant amount of processing power in addition to being computationally expensive, the cost of implementing this system can be large, especially when complex kernel functions and sizeable amounts of data are being used. This algorithm can also be sensitive to hyperparameters, causing it to require a great deal of tuning to reach optimal performance.

(Kontsewaya, 2021)study, evaluates the effectiveness of various machine learning methods. Methods such as Naive Bayes, Decision Tree, Random Forest, and k-Nearest Neighbors (k-NN) are included. The author’s findings suggest the

classifiers Naïve Bayes and Random Forest have the capability to outperform the other models in areas such as precision, F1-score, recall and accuracy. It was within the study that the Random Forest yielded the best results.

In the study by (Mohammed et al., 2013), an unsolicited bulk email classification system using Python-based machine learning techniques is proposed by the authors. Throughout the study, different machine learning algorithms are investigated. To evaluate the machine learning filters, various metrics are used. The results of the author's study indicate points towards the Random Forest classifier producing the best performance across all metrics.

Comparing the two studies, both arrive at similar conclusions. The Random Forest classifier demonstrated a superior performance in contrast to the other machine learning filters. Although the result was identical, the second-best filter varied between studies, with SVM being the runner up in (Mohammed et al., 2013) and Naive Bayes in (Kontsewaya, 2021).

The capability of adapting to new forms of spam in a short period of time whilst being accurate, is an advantageous point of machine learning-based filters. When new forms of spam appear, the filter is able to analyse, identify and then block these messages. There is no need for rule-based criteria or manual intervention. Factors such as the sensitivity of the filter and the priority of specific messages can also be customised for the users demands. However, a significant computational resource is required for machine learning-based filters to be trained and work effectively. This may present an issue for individuals or smaller organisations. Furthermore, personalized or highly targeted forms of spam (e.g. spear-phishing attacks) that can be modified to avoid the spam filter may not be detectable.

2.2.5 Deep Learning

Deep learning is a form of machine learning that utilizes artificial neural networks in order to learn and create accurate predictions from datasets. It is designed to imitate the process the human brain uses to process information. This is achieved through the use of interconnected layers of artificial neurons to extract features from the input data, aiding the creation of prediction using those features as the basis.

For email spam filtering, deep learning can be employed for the extraction of features and patterns for email data automatically, using them to classify the messages into non-spam or spam. This functionality can be remarkably beneficial for the detection of new and evolving types of spam which other filters, such as traditional machine learning filters and rule-based filters, may not be able to identify.

Although the algorithm tends to require a substantial amount of data for training causing it to be computationally intensive to operate, it can be exceptionally effective when it needs to learn complex relationships between the features of an email in correspondence to its classification. Deep learning can also surpass the performances of other filters after an adequate amount of training.

A commonly used type of deep learning is the Convolutional Neural Network (CNN). This types design enables it to use a grid-like structure to process data, inclusive of text and images, and extract features from data using convolutional layers (Yamashita et al., 2018). CNNs can be trained to extract relevant features from metadata and text residing in emails.

Recurrent Neural Networks (RNN) is another deep learning algorithm that can be utilized for email spam filtering. This algorithm is designed to process sequences

of data (text data and time-series) and use feedback loops to keep internal state, and capture dependencies between elements in within the sequence (DiPietro and Hager, 2020). Recurrent Neural Networks can be trained to process the text within the message a single sentence or word at a time, in addition to capturing dependencies amongst words or sentences, whilst maintaining context using its internal state.

(AbdulNabi and Yaseen, 2021) explores the use of various deep learning techniques including Convolutional Neural Networks (CNN) Long Short-Term Memory (LSTM) and a combination of both (LCNN). A comparison between these methods and other machine learning made, with the results showing LCNN outperforming the other models.

In contrast, (Alom, Carminati and Ferrari, 2020) introduces a deep learning model called the Stacked Autoencoder (SAE) in order to detect spam on twitter using the semantic features of Twitter messages. The model contains multiple layers of autoencoders, with each layer having been trained to reconstruct its input. After the training's completion, features from the input data is extracted by the model. The extracted features are then used to classify tweets as ham or spam. Within the study, the model's performance is compared with traditional machine learning algorithms. This comparison resulted in the SAE model attaining a higher F1-score and rate of accuracy than the other methods.

Both studies showcased the potential of deep learning techniques in detecting spam. These findings suggest that deep learning models are proficient enough to perform better than machine learning algorithms.

An advantage of Deep learning is its capability of accomplishing high accuracy rates and it can be trained for the identification of complex patterns within

emails. However, similar to other machine learning techniques, deep learning algorithms can be computationally expensive and require significant processing power. The method can also require a great amount of data in order to train the algorithm, with the possibility of the model being overfit to the input data.

2.2.6 Challenges

A main challenge in the field is the ever-evolving nature of spam. New techniques are continuously being created by spammers in an attempt to bypass spam filters in place. Due to this spam filters, are required to be updated on a regular basis to keep us with these changes. Additionally, there is the challenge of the trade-off between filtering out spam and allowing legitimate emails to pass through. If the filter is to lenient, spam may potentially pass through, whilst on the other hand, a strict filter can result in legitimate emails being blocked.

2.2.7 Recent Developments and Trends

Recently in the field of email spam filter technology, there has been a development with the use of artificial intelligence, deep learning algorithms and natural language processing. Using these technologies, the efficiency and accuracy of email spam filters can be refined and improved, formulating better filters. Cloud-based email spam filters are being used employed as it offers reliability and scalability, in addition to being cost-effective. Moreover, other forms of security measures are being merged, such as firewalls and antivirus software, with spam filters, resulting in a fortified guard against different forms of spam.

2.2.8 Conclusion

Email spam filters are a necessity for the protection and management of email inboxes from security threats. There are a variety of applicable spam filters that can be integrated for mitigation. Whilst filters such as rule-based filtering, blacklist filtering and whitelist filtering can be an effective tool for handling email spam, they should be used in conjunction with machine learning-based filters to create a higher level of precise spam detection. These filters should also be reviewed on a regular basis to verify whether or not legitimate emails have been delayed or blocked. Additionally, machine learning-based filters are an important tool to combat the issue of email spam. As machine learning algorithms continue to grow, this method of filtering will become even more effective. These algorithms should still be used with a combination of content-based filtering and rule-based filtering to ensure an all-inclusive approach to spam detection. The performance of machine learning-based algorithms also needs to be regularly evaluated to confirm the needs of the organization and the user are being met. With the consistently developing nature of email spam, it is vital that the correct measures are put in place for optimal security.

2.3 Case Study

2.3.1 Spam Filtering in a Small Business

The author's case study on spam filtering in a small business environment provides an in-depth analysis of the challenges and potential solutions for effective email filtering in the context of smaller organizations. Outlined within the study are the effects of spam emails on when it comes to things such as security, productivity and overall business operations. Additionally, the study

touches on unique needs and constraints faced by small businesses in caused by spam emails, such as budget constraints and limited resources.

The importance of implementing a multi-layered approach in spam filtering is emphasised, with the combination of various techniques such as machine learning algorithms, rule-based filtering, and user-generated white and black lists. Using a spam filtering strategy tailored to the business' needs, a higher level of protection can be reached by organizations.

In conclusion, valuable insight on how important implementing a comprehensive and comprehensive filtering strategy customized to meet the needs of a small business is given. Through the correct combination of different spam filtering techniques, employee education and updating the filtering process on a regular basis, small businesses can significantly improve their email security and productivity.

2.3.2 User Spam Filtering

During this the case study, the author investigates the effectiveness of user-based spam filtering techniques and system-wide methods. The study discovers user involvement can lead to more accurate and personalized filtering results, complementing system-wide filtering. Messages potentially overfitting or underfitting is presented as a challenge, in addition to inadequate user education.

The author recommends user education on email spam filter techniques being provided. User-friendly tools and a regular review being carried out, in addition to the user settings being updated is also suggested. The case study shows the importance of education needed to create a personalized and effective spam filtering experience whilst enhancing email security.

2.4 Existing Solutions

2.4.1 Introduction

This section aims to analyse and compare the existing solutions integrated within the system of the top three most popular email platforms in order to mitigate email spam.

2.4.2 Gmail

Gmail utilizes a great deal of rules to decipher whether or not an email is spam. Each rule is used for specified features, with particular statistical values attached to it which are dependent on the probability of the feature being spam (Dada et al., 2019). Using the weighted importance of each feature, an equation is then constructed. Each user has a sensitivity threshold in place, which is used to carry out a test. Depending on the result, the email receives a classification of either spam or non-spam. Additionally, state of the art machine learning-based algorithms such as neural networks and logistic regression are implemented by the platform to classify emails (Dada et al., 2019). Another algorithm used is optical character recognition (OCR) to protect users from potential image spam. Gmail is able to link together a collective of factors through the application of machine learning algorithms specially developed to rank massive sets of google search results, leading to the improvement of spam classification. The mail service also uses TensorFlow which is Google’s open-source machine learning framework and claimed by Google to be blocking over 100 million spam messages a day (Google Workspace Blog, n.d.). The application of this framework provides Google with the edge over other mailing platforms.

2.4.3 Outlook

Outlook spam filter uses machine learning algorithms in conjunction with other techniques for the identification of spam. As the platform employs a machine learning model, it learn from and adapts to new and unseen data over time.

2.4.4 Yahoo

Yahoo mail has its own spam algorithm that is used to detect spam messages. The email service utilizes “SpamGuard” technology (a machine learning algorithm), that is capable of automatically blocking emails from familiar spammers (Gregory, n.d.). Domains are also used to filter messages, in addition to a blend of other techniques. A list of the basic methods employed by the platform are email content, spam complaints from users and URL filtering.

2.4.5 Accuracy and Effectiveness

Although the statistics regarding the accuracy rates for the Yahoo and Outlook spam filters are not available online, Gmail shares the statistics for their spam filter. According to Gmail, 99.9% of spam is identified accurately, with the false positive rate at 0.05%. Additionally, Gmail reportedly have the best spam filter out of the three mail services.

2.4.6 Conclusion

In conclusion, all three spam filters have highly effective email spam filters. They all incorporate the key elements and factors required to form a protective wall of security. The mesh of techniques to combat and cover a wide range of types of spam creates a state-of-the-art system to mitigate threats. However, the Gmail spam filter is a cut above the rest. The main difference that sets it apart is the

utilization of TensorFlow within the system. This machine learning-based technique boosts the number of detected spam filters by a substantial amount, providing a higher level of defence.

CHAPTER 3

New Ideas

3.1 Introduction

The purpose of this project is to build a highly accurate spam filter through the adoption of various methods. As there will be a high workload for this project, it is important to formulate an efficient plan to reduce complications. Throughout this section, the proposed solution will be outlined. A detailed analysis of the project requirements which have been influenced by previous research will be included. Diagrams will also be used to display the structural design chosen for the project.

Following the previous chapter, a few of the researched techniques will be used to for the production of the spam filter. A selection of the pre-processing functions will be implemented in conjunction with a deep learning model. The deep learning model that will be used is TensorFlow. This model has been chosen as none of the examples in the review had administered it to a program to be tested. Gmail also utilizes this deep learning model, making it an interesting object for observation.

3.2 Aims and Objectives

3.2.1 Aims

The primary goal of this project is to develop an email spam filter capable of successfully classifying spam and non-spam emails. Using machine learning and

various filtering techniques, the filter should be capable of using a dataset learn the characteristics of that indicate spam messages through training. The model will then evaluate whether the message is legitimate or not.

3.2.2 Objectives

The following is a list of identified objectives:

- To use the deep learning models built with TensorFlow to identify patterns and features to distinguish legitimate and spam messages
- The create a model capable of learning from a dataset of emails that have already been classified to improve its accuracy and precision
- To improve the performance of the filter through feature engineering, data pre-processing and model optimization.
- The filters performance should be evaluated using metrics such as loss, accuracy and Area Under the Curve (AUC)
- The training results should be displayed

3.3 Project Planning

In order for any project to be successful, there must be a solid plan to create a foundation for the process. Effective project planning develops clear understanding of the scope. This can assist the prevention of delays and various other unpredictable factors that can potentially lead to the project's failure. The process to construct the project which will followed is outlined below.

3.3.1 Dataset collection

Before the spam filter is created, a dataset of spam and non-spam emails is required. The dataset will contain a substantial number of sample emails,

allowing a higher level of training to be achieved, increasing the accuracy. A publicly available dataset will be selected from a reliable source.

3.3.2 Data pre-processing

The data is required to be pre-processed before the model can be trained. Pre-processing is conversion of the raw text data into a useable format for the machine learning model (TensorFlow). The steps taken to carry out pre-processing are below:

Stop word removal: This removes common words such as “the”, “is”, “and”, “of” and so on. The removal of words that do not contribute an ample amount of meaning to the text reduces the performance of the model through the reduction of the number of features within the dataset.

Subject removal: The word subject will be removed from the email text. It will be replaced with an empty string.

Punctuation removal: Any punctuation characters (e.g. ?,’,/,.”) will be removed from the dataset.

Create a word cloud: The word cloud will be a visualization of the most common words within each classification.

Split the dataset: Divide the dataset into sets of data to be trained and tested.

Tokenization: During tokenization, the text is broken into words or tokens, allowing data to be processed more efficiently by the machine learning model.

3.3.3 Model Development

After the pre-processing has been carried out, TensorFlow and Keras will be used to create a neural network. There will be 4 layers consisting of input layer, the embedding, flatten and dense output layers. The embedding layer converts inputs into a continuous vector representation. Afterwards, the flatten layer will reshape the embedding layers output into a 1D tensor, representing the data as an array of numbers. The dense output layer will use a sigmoid activation function to map the output of the layer to a probability value. Next, the model will then be complied. To configure the model, adam optimizer, binary cross-entrophy and evaluation metrics will be used. Afterwards, the model will be trained with the dataset.

3.3.4 Model Evaluation

An evaluation will be carried out based on the training dataset. This will assess the model's performance by calculating the loss and metrics such as AUC and accuracy. The model's accuracy and loss will also be plotted as a graph.

3.3.5 Prediction on unseen data

A sample of new email data which the model has not seen before will be created. The new data will be put through the same pre-processing steps used previously on the dataset (removing stopwords, the subject and punctuation). Predications will then be made by the trained model, determining the probability of the unseen data being a spam or non-spam message.

3.4 Risks

The identification and assessment of potential project risks are a fundamental step of the process. Planning for these scenarios can aid their mitigation if they were to take place, resulting in the project remaining on schedule. This sub-section contains a few risks that could negatively impact the project if they were to occur and actions to take to mitigate them.

3.4.1 Risk Chart

Risk	**Severity**	**Likelihood**	**Risk Impact**	**Recommended action(s)**
Technological faults involving indispensable tools (e.g. Laptop or Wi-Fi)	Manageable	Possible	Medium/High	The university's library can be utilized
Loss/corruption of code	Detrimental	Possible	High	Will ensure code is regularly backed up
Laptop stolen	Detrimental	Not likely	High	The library computers can be used, and data should be regularly backed up
Project scope proves to be too intricate for current abilities	Acceptable	Not likely	Low	The project will be overviewed on a regular basis
Student or supervisor is absent for a	Manageable	Possible	Medium	Communication will be held regularly so prolonged periods

prolonged period of time				would not have a crucial effect
Potential lack of research for the project	Manageable	Possible	Medium	Ensure topic areas are researched thoroughly
Loss of report/documentation	Detrimental	Possible	High	Documents will be backed up regularly and saved to one drive, so it saves automatically
The schedule for the project and projected time tasks should take is longer than expected	Detrimental	Possible	High	The Gantt chart will be used to keep track of progress and along with any potential situational changes required to be managed

3.4.2 Project scope proves to be too intricate for current abilities

Expending excessive effort on specific areas without the ability to complete the task can waste time. In the case the task is too complicated, it may be simplified, or the intricate tasks will be removed.

3.4.3 Laptop stolen/lost and loss of documentation or code

Scenarios such as a device being misplaced or stolen are common occurrences, as well as hardware malfunctions and viruses. If this were to occur, there are computers available in on the university campus and at libraries. Additionally,

any data will be backed up on a regular basis, allowing it to be accessed through online services.

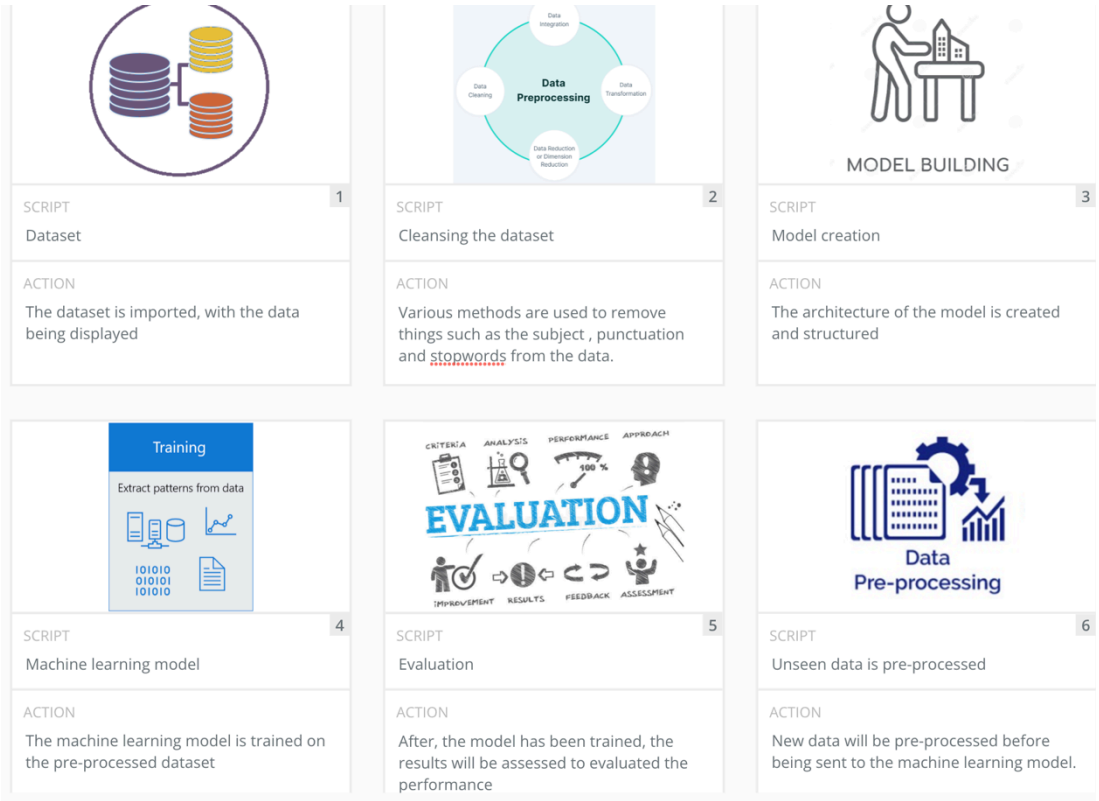
3.4.4 The schedule for the time tasks should take is longer than expected

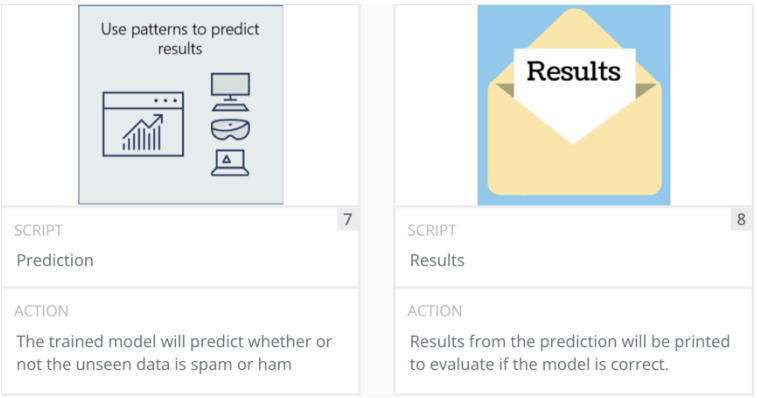
As the project involves multiple stages of development, which may lead to unpredictable situations. However, mitigating this risk is a feasible task. The employment of correct time management, in conjunction with thorough and realistic elements of planning can prevent this scenario.

3.5 Design

The design phase communicates the project vision helps ensure the it is scalable and efficient. This sub-section will act as the beginning of the creation of a more tangible form to the project.

3.5.1 Story Board

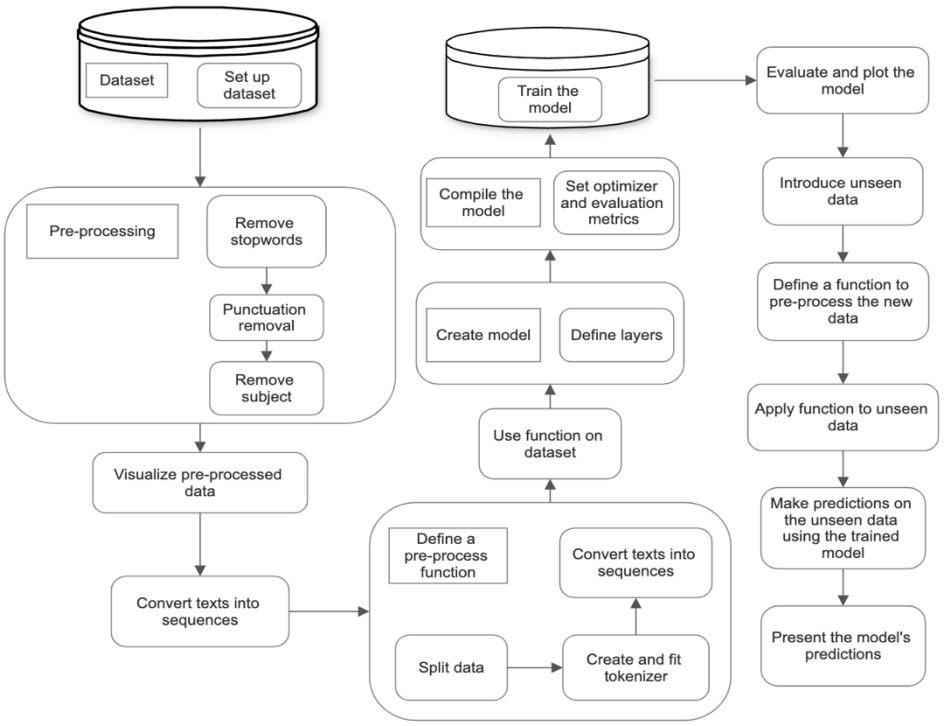




The storyboard illustrates the key stages of the spam and ham email classification process. It highlights the importation of the data before it is then cleansed using pre-processing. Subsequently, the model is built and complied so it can be train using the dataset. Evaluation takes place after the training has

been completed. Next, new data is pre-processed to so the machine learning model can be tested. Finally, the results are displayed after the prediction.

3.5.2 Block Diagram



The block diagram exhibits a clear, concise and comprehensive overview of the entire spam and ham email classification process. Each stage is involved in building and deploying the machine learning model is showcased. It begins with data being imported and then visualized so the distribution of ham and spam emails can be seen. Then, it moves onto the pre-processing steps, which cleanses the data, so it is prepared for training. The diagram also highlights the data being split before the tokenization and sequencing takes place, transforming the data into a suitable format for the machine learning model. These will all be integrated into a singular function. Next, the model's architecture is designed through the definition of layers, before it is complied with an optimizer and evaluation metrics. The machine learning model is then

trained using the special parameters that have been set. After, the model's evaluation, a new pre-process function is created to be used on the new data. When the data has been processed, the model makes prediction based on the training it has received.

CHAPTER 4

IMPLEMENTATION

4.1 Introduction

This chapter will outline the methods and process of the project's development. Selection choices such as the software and programming language will also be included.

The code design layout within the previous chapter will be used as a structural basis for the code. The features will also be integrated into the project.

4.2 Methodologies

Selecting a suitable methodology is vital as it affects the approach taken towards the project. In order to choose an appropriate methodology for the project development, the characteristics of the project were taken into account.

Email spam filters require continuous development and improvement. As the project progresses it may be necessary to return to areas of the code for improvement, meaning various areas of the code would need to be developed in a parallel and iterative format.

Selecting a suitable methodology is vital as it affects the approach taken towards the project. In order to choose an appropriate methodology for

the project development, the characteristics of the project were taken into account.

4.2.1 Agile Model

Email spam filters require continuous development and improvement. As the project progresses it may be necessary to return to areas of the code for improvement, meaning various areas of the code would need to be developed in a parallel and iterative format.

The Agile model is an iterative and incremental approach to software development. Unlike traditional software development models, which rely on detailed planning and documentation upfront, the Agile model is based on the idea of rapid prototyping and continuous improvement. The model is centred around a series of short development cycles, called sprints. Each sprint lasts one to four weeks. The Agile model spotlights close collaboration between the development team and the customer/end-user. There is a constant flow of communication, ensuring the software meets the customer's needs. Flexibility and adaptability are emphasized features within the model, in addition the aptness of being able to make changes to the software based on feedback. Overall, the model is a collaborative, and a flexible approach to software development that pushes the idea of continuous improvement and prototyping.

As email spam filtering needs a high degree of accuracy and effectiveness, the agile model's focus on continuous improvement and prototyping ensures that the software is constantly being refined and improved. This can lead to a more refined filter.

4.3 Development

This section outlines the steps taken during the creation of the project The development took place on the platform Jupyter Labs, using python as the main programming language.

4.3.1 Setting up the dataset

Before beginning to develop the project, the correct imports had to be downloaded and inserted.

The first step was to insert the dataset into the pandas data frame. After the data frame had been successfully inserted, a code to plot a graph displaying the amount of spam and non-spam emails was plot. The import "seaborn" was used as "sns" to create a statistical graph. Additionally, the axis' were labelled with matplotlib.

4.3.2 Pre-processing

As explained in section 3, the next step was to filter the dataset during the pre-processing. The initial step was to remove the stop words from the data. Nltk, a natural language toolkit was imported from the python library. In the code above, the "stop = stopwords.words('english')" removes the words which are in line with the stopwords dictionary from the dataset. The remaining words within the dataset are then stored as important if they are not removed by the "stop" function.

Next in the process is the removal of the subject from the data. This implementation was relatively simple. Str.replace was used to replace the subject string in the dataframe column.

4.3.3 Developing the Model

To begin the model's development, the model.compile() function is used to configure the model for training using the 'adam' optimizer, 'binary_crossentropy' loss, accuracy and AUC (Area Under the Curve) metrics.

The model is trained for a specific number of epochs, which are the iterations over the entire dataset, the model.fit() function was implemented. The number of epochs was set to 10, with the batch size set to 300. These were made variable which are passed into the model.fit. The valdation_data parameter has been set to the test dataset, meaning after each epoch the model's performance will be evaluated.

An EarlyStopping callback was also added. The purpose of the callback is to monitor the val_loss (validation loss) and then stop the training in the scenario there is no improvement after 4 consecutive epochs. Additionally, the model's weights from the best performing epoch is restored after the training is complete using 'restore_best_weights'.

It was found that sometimes when attempting to run the code, the shape of the input layer changed. The cause of this is still unknown. However, to fix this issue the code for how the layer is passed into the model for training was changed. Before, the integer for the number was pre-defined

in the layer, but after the modification the variable max_seq_length was inserted. This variable contains the maximum length of the token sequences generated from the raw training texts. Using this variable, the possibility of the input shape error was removed. This was feasible due to the agile method, allowing changes to the code when needed.

4.3.4 Model Evaluation System

To get a better view of the model performance, charts were plotted to display the accuracy and loss rate of the model. The validation accuracy and validation loss are plotted for each epoch throughout the training process using model_accuracy. Visualizing the graphs performance helps determine whether or not the model is overfitting or underfitting. This was achieved due to an implementation when the code to train the model was being built. At the start of the code, a variable named history was defined. Using this variable, the training history could be stored and called upon later. The history was then attributed to a graph, with a title being given and the axis labelled.

4.3.5 Predicting Based on Unseen Data

After a new sample of data was created for the trained model to be tested, the preprocess_unseen_data function was implemented as a method to pre-process unseen data before the model makes predictions. For this function to successfully operate, a copy of the unseen dataframe was produced in order to avoid the modification of the original data. Previous methods used to carry out the pre-processing were then called and applied to the unseen data, putting it through the same steps as the training data.

```
# Preprocess unseen data
def preprocess_unseen_data(unseen_data, tokenizer):
    unseen_data = unseen_data.copy()
    unseen_data['text'] = unseen_data['text'].apply(lambda text: remove(text))
    unseen_data['text'] = unseen_data['text'].str.replace('subject', '')
    unseen_data['text'] = unseen_data['text'].apply(lambda x: puncRemoval(x))

    new_sequences = get_sequences(unseen_data['text'], tokenizer, train=False, max_seq_length=train_text.shape[1])
    return new_sequences
```

The tokenizer that was created at an earlier phase in the code, is then fitted onto the training data 'train_text_raw'. The tokenizer being fitted onto the unseen data shows the tokenizer can be fitted onto both sets of data and use the same word-to-integer mapping. As the model can only recognize words seen during training it is important the tokenizer remains consistent across datasets.

Afterwards, the preprocess_unseen_data function is called to pre-process the unseen data the tokenizer has been fitted with. To re-create sequences with the same length, get_sequences is utilized, turning the pre-processed text into sequences of integers. This process creates a consistency with the type of data given to the model. The variable predictions is then used to contain the probability values for each input sequence within the unseen data.

In order for the probabilities to be converted into binary label, a threshold is set. The threshold states that if the number is greater than 0.5 then the decided label is spam. Otherwise it is classified as a legitimate message. As only raw text messages are contained within 'unseen_data', a new column named predictions is added to the dataframe. Binary predictions are then added to the unseen data before the dataframe is printed, causing the original text and predictions to be displayed.

CHAPTER 5

RESULTS / DISCUSSION

5.1 Introduction

To determine whether the spam filter is fully functional and meets the aims and objectives outlines in section 3, it is required to be tested and evaluated. To achieve this, a testing plan has been created, with an evaluation of the results being included. The method of testing covers the functional requirements.

5.2 Testing Plan

The testing plan has been crafted to ensure the robustness and effectiveness of the implemented deep learning model.

Requirement	**Result**
Dataset Validation	
Pre-processing	
Sequences and Tokenization	
Model Creation	
Model Training	
Model Evaluation	

Unseen Data Prediction	

5.3 Testing

5.3.1 Dataset Validation

- Check if the dataset is loaded correctly.
- Check if the dataset has the expected columns.

	Unnamed: 0	label	text	label_num
0	605	ham	Subject: enron methanol ; meter # : 988291\r\n...	0
1	2349	ham	Subject: hpl nom for january 9 , 2001\r\n(see...	0
2	3624	ham	Subject: neon retreat\r\nho ho ho , we ' re ar...	0
3	4685	spam	Subject: photoshop , windows , office . cheap ...	1
4	2030	ham	Subject: re : indian springs\r\nthis deal is t...	0
...	...	...	...	...
5166	1518	ham	Subject: put the 10 on the ft\r\nthe transport...	0
5167	404	ham	Subject: 3 / 4 / 2000 and following noms\r\nhp...	0
5168	2933	ham	Subject: calpine daily gas nomination\r\n>\r\n...	0
5169	1409	ham	Subject: industrial worksheets for august 2000...	0
5170	4807	spam	Subject: important online banking alert\r\ndea...	1

The dataset is adequately displayed with all the expected columns.

5.3.2 Pre-processing

- Use a sample string to test the remove function
- Test the puncRemoval function with a sample string.
- Use a small dataset to test word cloud

- Test the pre-process function with a small dataset.
- Verify the output shapes and value distributions.

All the pre-processing features were tested and carried out their required functions successfully. This means the program is capable of transforming raw text data into a more structured and compatible format to be used by the machine learning model. When these functions carry out their designated tasks, the text can be standardized, causing the most relevant features to be highlighted and the model's performance to be enhanced.

5.3.3 Sequences and Tokenization

- Test the get_sequences function with a sample string.
- Check if sequences are generated correctly.

When given a sample input string, the function utilizes the pre-trained tokenizer to convert the text into a sequence of integers. The functions success makes it evident it is capable of handling training and test data, dynamically adjusting the sequence length based on the input dataset. Furthermore, all the integration of padding techniques provides all sequences will the same length. As a result, data can be efficiently processed in batches.

5.3.4 Model Creation

- Check if the model is created without errors.
- Verify the model architecture.

```
Model: "model_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_5 (InputLayer)        [(None, 3472)]            0

 embedding_4 (Embedding)     (None, 3472, 64)          1920000

 flatten_4 (Flatten)         (None, 222208)            0

 dense_4 (Dense)             (None, 1)                 222209

=================================================================
Total params: 2,142,209
Trainable params: 2,142,209
Non-trainable params: 0
_________________________________________________________________
None
```

The architecture for the model is created without and errors and the order of the layers are clearly displayed. Consisting of an input layer, embedding layer, flatten layer and a dense output layer, with a sigmoid function, the architecture is effective at learning and extracting important features from the text. Transforming sparse textual data into dense vectors, the embedding layer enabling the model to capture the relationships between words and their context. The dense output layer with the sigmoid activation function allows clear binary classification of emails as it ensures the final output is a probability between 0 and 1.

5.3.5 Model Training

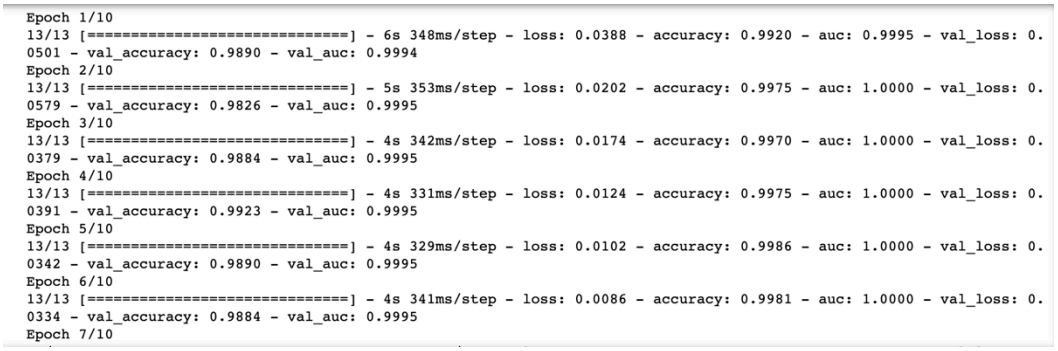
- Train the model using pre-processed version of the dataset.
- Check if the model trains without errors.

```
Epoch 1/10
13/13 [==============================] - 7s 484ms/step - loss: 0.8450 - accuracy: 0.6400 - auc: 0.5287 - val_loss: 0.
7246 - val_accuracy: 0.2906 - val_auc: 0.8912
Epoch 2/10
13/13 [==============================] - 6s 461ms/step - loss: 0.6174 - accuracy: 0.7110 - auc: 0.5976 - val_loss: 0.
5391 - val_accuracy: 0.7101 - val_auc: 0.9763
Epoch 3/10
 8/13 [=================>............] - ETA: 1s - loss: 0.5177 - accuracy: 0.7237 - auc: 0.8279
```

No errors occurred whilst the model was being trained using pre-processed errors. The training of the algorithm being successful outlines its robust architecture and efficiency of the chosen pre-processing and training techniques. A steady improvement in the model's ability to accurately identify spam and ham emails was demonstrated over the course of the training. This is evident by the increasing accuracy metrics and decreasing loss. Techniques such as sequence padding, tokenization, and early stopping callback ensured that a balance between learning complex patterns and overfitting can be accomplished. The consistent tuning of the model parameters and updating its weights allowed the training process teach the model correct classifications through email characteristics.

5.3.6 Model Evaluation

- Check if the model performs as intended.



The model's architecture, which includes an embedding layer and a fully connected dense layer with a sigmoid activation function, has proven to be effective in capturing the underlying patterns within the data.

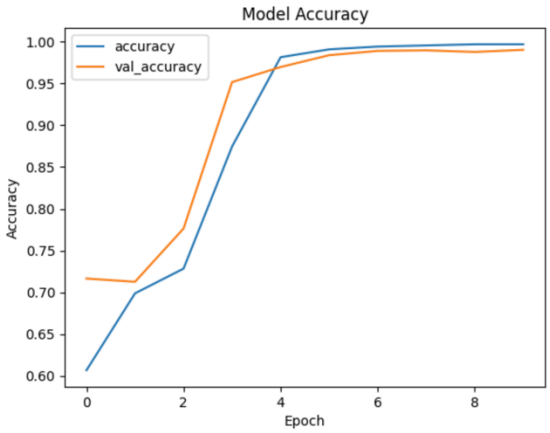
As shown is the figures above, the model can successfully use the dataset to train and learn to classify spam and ham messages. The high accuracy and AUC scores accomplished during the training and testing show the model can be attributed to its ability to perform as intended. Furthermore, the pre-processing steps used to remove stop words, punctuation and tokenize the text have aided the model’s ability to focus on more on the most relevant features within text.
The model's architecture, which includes an embedding layer and a fully connected dense layer with a sigmoid activation function, has proven to be effective in capturing the underlying patterns within the data.

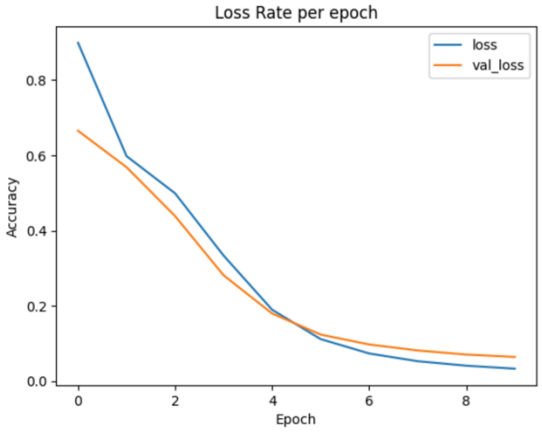
- Evaluate the model on using test data.

```
49/49 [==============================] - 0s 10ms/step - loss: 0.0293 - accuracy: 0.9897 - auc: 0.9995
Test Loss:  0.02925085835158825
Test Accuracy:  0.9896907210350037
Test AUC:  0.9995407462120056
```

The output for the evaluation of the model on the test data shows it has a high accuracy with a low loss. These results display the model is learning and adapting to the provided dataset.

- Test the model_accuracy and loss_comparison functions.





Both functions outputting the requested statistics for the model's shows the model is capable of differentiating between spam and ham emails. These functions provide valuable insights into the model's learning process by plotting the training, validation accuracy and loss rates over a multiple epochs. The 'model_accuracy' function particularly portrays the model's success in classifying emails correctly. This is a further indication the model is not overfitting to data and can consistently detect spam emails. The loss_comparison function highlights the optimization process, showing a steady decrease in training and validation losses. This signifies that the model is effectively learning to minimize errors in its prediction. The execution and visualization of these functions exhibits the reliability of the model in conjunction with its intended purpose being met.

5.3.7 Unseen Data Prediction

- Test the preprocess_unseen_data function with a few sample strings.
- Make predictions on the pre-processed unseen data.
- Verify if the predictions are in the expected format.

	text	predictions
0	You won a lottery! Claim your prize now!	1
1	Meeting rescheduled to tomorrow at 3 pm.	0
2	Dentist appointmtnt reminder for Tuesday 15th ...	0
3	YOU STILL HAVE NOT CLAIMED YOUR FREE IPHONE 13...	1
4	Are interested in joining our special program?	1

Various sample strings were used to test the preprocess_unseen_data. The pre-processed unseen data was then passed through the model and returned. The predications were printed alongside the text in the correct binary format. The model is adept at accurately classifying the majority of unseen data, presenting its capacity to rationalize new instances. This result shows the model has successfully learned the underlying patterns within the data. Moreover, the unseen data was passed through the same pre-processing steps, providing fair comparisons between performances.

5.4 Evaluation

The testing indicates the project has been successful in achieving its objectives. By employing a train-test split, the plan facilitated an unbiased assessment of the model's performance on unseen data. This approach helped avoid overfitting and provided a more accurate representation of the model's ability to generalize to new examples. Additionally, the use of performance metrics, such as accuracy and area under the curve (AUC), enabled a comprehensive evaluation of the model's performance. These metrics provided valuable insights into the model's

classification capabilities, revealing its strengths and potential areas for improvement. Overall, the testing plan effectively guided the development and assessment of the email spam filtering model, ensuring its reliability and suitability for real-world applications.

CHAPTER 6

CONCLUSIONS / FUTURE WORK

6.1 Conclusions

The aim of this project was to develop an email spam filter capable of analysing and detecting unsolicited emails which may cause harm. This project was successful because the email spam filter was successfully created and proven to be functional. The filter demonstrates an effective approach using a deep learning model. Furthermore, the application of the trained model to classify unseen email data, showcases its potential to be applied in real-world scenarios.

The project has indicated the value of employing a comprehensive and suited approach to spam emails. Throughout the project, various techniques and methodologies have been explored and implemented. This ranges from rule-based filters to deep learning techniques. By combining these diverse methods, a higher level of accuracy has been achieved in the distinguishment between legitimate and spam emails, minimizing false positives and negatives.

Moreover, the significance of a multi-layered strategy capable of adapting to the specific needs and constraints of different organizations, such as small businesses with limited resources. By incorporating a customized combination of spam filtering techniques, organizations can find a balance between cost-effectiveness and optimal protection against unwanted emails and their associated risks.

The critical role played by employee education and training were highlighted in the case studies. The human factor remains an essential component of email security, it is important for employees to be knowledgeable in the subject area. Employees containing the skills to identify and handle spam emails in an effective manner can increase an organizations security against cyber threats.

Furthermore, the significance of continuous monitoring, evaluation, and improvement of the spam filtering system has been outlined. As the tactics and techniques applied by spammers keeps evolving, it is vital to stay up to date with the latest trends and advancements in the spam filtering technology field.

Overall, the email spam filter project has successfully demonstrated the value of a comprehensive, multi-faceted and adaptable approach to mitigate spam emails.

6.2 Future work

Future work is a collection of proposed features that can be used to expand the project .

Advanced deep learning techniques

More advanced techniques for can be employed, refining the model. These architectures could potentially capture more complex patterns in the text and improve the filter, refining the model.

Ensemble models

The combination of multiple machine learning models or algorithms can lead to an overall better performance. Ensemble techniques such as bagging, stacking, or boosting create more robust and accurate spam filtering systems.

Multilingual support

The current model is designed for English text. In the future models can be developed to support multiple languages, assisting users across the globe.

Additional features

As the current model relies primarily on text data, future work can explore the incorporation of other features to improve performance. Some of these are user behaviour, metadata, sender reputation, email.

Real-time spam filtering

Users can be provided with an efficient email experience with the investigation of the development of real-time spam filtering systems that are able to classify emails as they are received.

Transfer learning

Using pre-trained models such as BERT, GPT or RoBERTa the performance can improve. This will transfer the knowledge from large-scale pre-training taking place on diverse text corpora.

6.3 Legal, Social, Ethical and Professional Issues

Legal Issue	**Description**
Equality Act 2010	The Equality Act was created in order to legally protect people from discrimination throughout society. In relevance to the project, it should be

	equally accessible by anyone, regardless of religion, sex or abilities.
Computer Misuse Act 1990	The creation of this act put in place a law to control individuals' ability to access data lawfully through a computer system. The project is required to have no underlying functions which bypass access to user's data unknowingly.
GDPR and Data Protection Act 1998	As a step to secure personal data being stored on computers the GDPR and Data Protection Act were put in place. All data collected from the project is required to be kept confidential.

Social Issue	**Description**
Confidentiality	As stated above, confidentiality is an important issue not just socially but legally too. Any collected data from the project should remain confidential and secured.
Accessibility	Accessibility is vital as multiple people with issues such as visual impairment. The formatting of the project and program will be in the parameters of a widely accessible format.

Ethical Issue	**Description**

Deceptive Advertisement	Displaying a false perception on what the program will provide and aims to do is against ethics as it is deceiving the user. The functions of the program will be clearly outlined and if they cannot be met, the outlined functions will be altered.
Data Theft	Using the program to steal the user's data without their knowledge and then use it for other purposes is extremely unethical and against the legal issues spoken about above. Any data collected will be with the user's consent and kept secure.

BCS Code of Conduct Rules	**Description**
You make IT for everyone	Discrimination regardless of the form it is presented is never to be tolerated. The project should be sculpted into a form which is accessible by everyone.
Show what you know, learn what you don't	Indicating towards the legal issues, all legislation rules are mandatory to be followed. "Ensure that you have the knowledge and understanding of legislation and that you comply with such legislation, in carrying out your professional responsibilities" (BCS, 2022). Additionally, any attempt to provide a service should be within the individuals skillset. All aspects of professional competence and legal

	aspects will be put into consideration throughout the project.
Respect the organisation or individual you work for	The creator is responsible for all work produced and should always work with the user's best interest at all times. In accordance to data protection, confidential information should not be disclosed without the required legislation or authority. Just as mentioned in the social issues segment, confidentiality is a crucial aspect in the industry and is a rule which will be maintained during the projects course.
Keep IT real. Keep IT professional. Pass IT on	This point solidifies the ideology of upholding professional standards as being a member of the BCS makes you an embodiment and representative of the industry. Respect should also be present within relationships, in addition to integrity. This mindset will be upheld throughout interactions will supervisors and the project.

6.4 Synoptic Reflections

Throughout the course of the project, several key insights and skills were acquired. The intricacies of natural language processing (NLP), including data pre-processing techniques such as tokenization, stop-word removal, and

punctuation elimination were studied and put to practice. The project also highlighted the importance of exploratory data analysis, visualizations and understanding the distribution of the dataset. Using graphs like count plots and word cloud helped grasp the differences between spam and ham emails. The use of TensorFlow's Keras API demonstrated the ease and flexibility of working with deep learning models for NLP tasks. The project provided hands-on experience in building, training, and evaluating neural networks.

This email spam filtering project has been an invaluable learning experience. The newfound comprehensive understanding of NLP, deep learning techniques, model evaluation, and deployment have broadened my skillset and knowledge in the cyber security field. Furthermore, they will be applicable in the future jobs as I use it as a foundation.

Over the course, I have taken in a plethora of new information. The diverse modules acted as a network of information for me to build on. An understanding of how various areas of not just cyber security but also computer science are connected, was refined as I attended lectures and completed coursework. This encompassed areas such as cryptography, ethical hacking, risk management and network security. As a result of learning about various types of cyber-attacks and how to defend against them, the course has provided me with knowledge to take with me and apply during employment.

REFERENCES

AbdulNabi, I. and Yaseen, Q. (2021). Spam Email Detection Using Deep Learning Techniques. *Procedia Computer Science*, 184, pp.853–858. doi:https://doi.org/10.1016/j.procs.2021.03.107.

Alom, Z., Carminati, B. and Ferrari, E. (2020). A deep learning model for Twitter spam detection. *Online Social Networks and Media*, 18, p.100079. doi:https://doi.org/10.1016/j.osnem.2020.100079.

Awad, W.A. (2011). Machine Learning Methods for Spam E-Mail Classification. *International Journal of Computer Science and Information Technology*, 3(1), pp.173–184. doi:https://doi.org/10.5121/ijcsit.2011.3112.

Cinelli, M., Sun, Y., Best, K., Heather, J.M., Reich-Zeliger, S., Shifrut, E., Friedman, N., Shawe-Taylor, J. and Chain, B. (2017). Feature selection using a one dimensional naïve Bayes' classifier increases the accuracy of support vector machine classification of CDR3 repertoires. *Bioinformatics*, p.btw771. doi:https://doi.org/10.1093/bioinformatics/btw771.

Dada, E.G., Bassi, J.S., Chiroma, H., Abdulhamid, S.M., Adetunmbi, A.O. and Ajibuwa, O.E. (2019). Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon*, 5(6), p.e01802. doi:https://doi.org/10.1016/j.heliyon.2019.e01802.

DiPietro, R. and Hager, G.D. (2020). Deep learning: RNNs and LSTM. *Handbook of Medical Image Computing and Computer Assisted Intervention*, pp.503–519. doi:https://doi.org/10.1016/b978-0-12-816176-0.00026-0.

Google Workspace Blog. (n.d.). *Spam does not bring us joy—ridding Gmail of 100 million more spam messages with TensorFlow*. [online] Available at: https://workspace.google.com/blog/product-announcements/ridding-gmail-of-100-million-more-spam-messages-with-tensorflow [Accessed 2 Feb. 2023].

Gregory, S. (n.d.). *How to Make Yahoo Spam Filter Work*. [online] Small Business - Chron.com. Available at: https://smallbusiness.chron.com/make-yahoo-spam-filter-work-59846.html [Accessed 5 Feb. 2023].

Hassan, M.A. and Mtetwa, N. (2018). *Feature Extraction and Classification of Spam Emails*. [online] IEEE Xplore. doi:https://doi.org/10.1109/ISCMI.2018.8703222.

Ismail, S.S.I., Mansour, R.F., Abd El-Aziz, R.M. and Taloba, A.I. (2022). Efficient E-Mail Spam Detection Strategy Using Genetic Decision Tree Processing with NLP Features. *Computational Intelligence and Neuroscience*, [online] 2022, p.e7710005. doi:https://doi.org/10.1155/2022/7710005.

Kontsewaya, Y., Antonov, E. and Artamonov, A. (2021). Evaluating the Effectiveness of Machine Learning Methods for Spam Detection. *Procedia Computer Science*, 190, pp.479–486. doi:https://doi.org/10.1016/j.procs.2021.06.056.

Kumar, N., Sonowal, S. and Nishant (2020). *Email Spam Detection Using Machine Learning Algorithms*. [online] IEEE Xplore. doi:https://doi.org/10.1109/ICIRCA48905.2020.9183098.

Ma, T.M., YAMAMORI, K. and Thida, A. (2020). *A Comparative Approach to Naïve Bayes Classifier and Support Vector Machine for Email Spam Classification*. [online] IEEE Xplore. doi:https://doi.org/10.1109/GCCE50665.2020.9291921.

Mohammed, S., Mohammed, O., Fiaidhi, J., Fong, S. and hoon Kim, T. (2013). *Classifying Unsolicited Bulk Email (UBE) using Python Machine Learning Techniques* .[Accessed 20 Apr. 2023].

Ramón Méndez Reboredo, J. (2005). *Tokenising, Stemming and Stopword Removal on Anti-spam Filtering Domain*. [online] Available at: https://www.researchgate.net/publication/221275087_Tokenising_Stemming_and_Stopword_Removal_on_Anti-spam_Filtering_Domain [Accessed 20 Apr. 2023].

Reddy Guda, S. (2022). *Evaluation of Email Spam Detection Techniques Evaluation of Email Spam Detection Techniques*. [online] Available at: https://repository.stcloudstate.edu/cgi/viewcontent.cgi?article=1169&context=msia_etds [Accessed 16 Dec. 2022].

Ruskanda, F. (2019). *Study on the Effect of Preprocessing Methods for Spam Email Detection*. [online] Available at: https://www.researchgate.net/publication/331961476_Study_on_the_Effect_of_Preprocessing_Methods_for_Spam_Email_Detection [Accessed 20 Apr. 2023].

Saidani, N., Adi, K. and Allili, M.S. (2020). A semantic-based classification approach for an enhanced spam detection. *Computers & Security*, [online] 94, p.101716. doi:https://doi.org/10.1016/j.cose.2020.101716.

Sharma, M., Scholar, P., Sharma, I. and Cse, H. (2018). A Survey of Email Spam Filtering Methods. [online] 7. Available at: https://core.ac.uk/download/pdf/234676898.pdf [Accessed 16 Dec. 2022].

Shrivastava, J.N. and Bindu, M.H. (2014). E-mail Spam Filtering Using Adaptive Genetic Algorithm. *International Journal of Intelligent Systems and Applications*, 6(2), pp.54–60. doi:https://doi.org/10.5815/ijisa.2014.02.07.

V.Christina, S.Karpagavalli and G.Suganya (2010). *Email Spam Filtering using Supervised Machine Learning Techniques* .

Yamashita, R., Nishio, M., Do, R.K.G. and Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, [online] 9(4), pp.611–629. doi:https://doi.org/10.1007/s13244-018-0639-9.

BIBLIOGRAPHY

Brownlee, J. (2019). *Your First Deep Learning Project in Python with Keras Step-By-Step*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/#:~:text=Keras%20is%20a%20powerful%20and [Accessed 15 Mar. 2023].

PyPI. (2019). *tensorflow*. [online] Available at: https://pypi.org/project/tensorflow/ [Accessed 13 Feb. 2023].

www.datacamp.com. (n.d.). *Python Word Clouds Tutorial: How to Create a Word Cloud*. [online] Available at: https://www.datacamp.com/tutorial/wordcloud-python [Accessed 5 Mar. 2023].

APPENDIX A