Name:	Sheikh Muhammad Ahmad
Student Id:	10637344
Module Title:	Applied Research Project
Module Code:	B9CY107
Module Supervisor:	Dr. Vivek Kshirsagar
Project Title:	Spam Classification Using Machine Learning and Deep Learning
Individual/Group:	Individual

SPAM CLASSIFICATION USING MACHINE LEARNING AND DEEP LEARNING

Declaration

Acknowledgement

‘I declare that this Applied Research Project that I have submitted to Dublin Business School for the award of MSc Cybersecurity is the result of my own investigations, except where otherwise stated, where it is clearly acknowledged by references. Furthermore, this work has not been submitted for any other degree.’

Signed: Sheikh Muhammad Ahmad

Student Number: 10637344

Date: 02-Jan-2024

Table of Contents

List of figures

List of Tables

Abstract

This research explores a comprehensive approach to refining spam classification accuracy by integrating traditional machine learning and advanced deep learning models. Our evaluation encompasses four diverse models—Adaboost, XGBoost, Long Short-Term Memory (LSTM), Feedforward Neural Network (FFN), and an innovative Transformer-CNN hybrid model. Notably, XGBoost emerges as the frontrunner, achieving a remarkable accuracy of 97.84%, closely trailed by Adaboost at 96.72%. The deep learning counterparts, LSTM and FFN, demonstrate competitive accuracies of 96.47% and 97.67%, respectively. Furthermore, the proposed Transformer-CNN hybrid model exhibits a commendable accuracy of 97.07%. This study underscores the pivotal role of amalgamating diverse machine learning paradigms, emphasizing the efficacy of hybrid models in significantly enhancing accuracy and overall performance in spam classification. The findings contribute valuable insights to the domain, showcasing the potential of a unified approach in fortifying email security and advancing the state-of-the-art in spam detection mechanisms.

1. Introduction

In an era dominated by digital communication, the ubiquitous nature of email makes it a primary vector for spam, posing substantial threats to user experience and data security. Traditional methods of spam detection often fall short in addressing the evolving tactics employed by spammers. This research presents a sophisticated strategy to enhance spam classification accuracy by integrating the strengths of both traditional machine learning and cutting-edge deep learning models. Our investigation involves the evaluation of four distinct models—Adaboost, XGBoost, Long Short-Term Memory (LSTM), Feedforward Neural Network (FFN), and a novel Transformer-CNN hybrid model. As email-borne threats continue to evolve in complexity, the amalgamation of diverse machine learning paradigms becomes imperative. This study not only contributes to the field of cybersecurity but also establishes a benchmark for the efficacy of hybrid models in fortifying email security, thereby addressing the critical need for advanced spam classification methodologies in contemporary digital communication landscapes.

1.1 Back Ground scope

In the ever-evolving landscape of digital communication, the pervasive issue of spam messages has become a persistent challenge, significantly impacting user experience and posing potential security risks. This paper delves into recent research endeavors that leverage machine learning (ML) and deep learning (DL) methods for the automatic classification of spam in both Short Message Service (SMS) and email communication channels. The insights drawn from a collection of studies contribute to the overarching goal of developing effective spam detection mechanisms and shed light on the multifaceted nature of this ongoing challenge.

The surge in spam messages across SMS and email platforms has necessitated the development of robust automatic classification techniques. Manual identification is impractical due to the sheer volume of messages, underscoring the significance of ML and DL algorithms in empowering communication platforms with autonomous spam detection capabilities. These technologies aim to discern between legitimate and spam messages, mitigating the impact of unsolicited communication on users.

Abayomi-Alli et al. (2022) address the challenge of SMS spam with a deep learning method, emphasizing the importance of context-specific solutions. Their study, conducted on an indigenous dataset, not only highlights the effectiveness of deep learning algorithms in discerning spam messages in SMS communication but also underscores the need for tailored approaches to account for the unique characteristics of different datasets. This contextual awareness is crucial for the development of spam detection systems that can adapt to diverse linguistic and cultural nuances.

In the realm of email communication, Mallampati and Hegde (2020) contribute a machine learning-based framework for email spam classification. Their work sheds light on challenges specific to email, such as the diverse nature of content and the need for nuanced classification models. Additionally, Sheneamer (2021) offers a comparative analysis of deep and traditional learning methods for email spam filtering, providing insights into the relative strengths and weaknesses of these approaches. The exploration of email-specific challenges further contributes to the broader understanding of effective spam detection methodologies.

Turning attention to phishing emails, Bagui et al. (2021) employ machine learning and deep learning techniques, placing emphasis on one-hot encoding as a method for feature representation. This approach showcases the diversity of methods within the broader field of spam classification, catering to the unique challenges posed by phishing attempts, which often involve sophisticated social engineering tactics. The study underscores the importance of staying ahead of evolving tactics employed by spammers and continuously refining classification models.

Expanding the research scope, Karasoy and Ballı (2022) focus on spam SMS detection in the Turkish language. Their study incorporates deep text analysis and deep learning methods, recognizing the language-specific nuances that influence the effectiveness of spam detection algorithms. This linguistic and cultural awareness highlights the need for region-specific models that account for variations in language use, slang, and contextual cues.

The research landscape reveals both challenges and opportunities in the ongoing quest to combat spam. Challenges include the dynamic nature of spam, the necessity for diverse and representative datasets, and the evolving tactics employed by spammers to evade detection. Opportunities, on the other hand, lie in the continuous advancements in ML and DL, offering the potential for more accurate and adaptive spam classification systems. Ongoing collaboration among researchers, industry stakeholders, and policymakers is essential to address these challenges collectively and develop comprehensive strategies to tackle the spam epidemic.

The highlighted research underscores the urgency of addressing the spam epidemic in digital communication. The diverse range of studies emphasizes the need for context-aware solutions, the exploration of language-specific challenges, and the ongoing evolution of techniques to stay ahead of spammers. As technology continues to advance, this body of research provides a foundation for the development of more robust and adaptive systems to safeguard communication channels from the onslaught of unwanted messages. The collaborative efforts of researchers across different domains contribute to the collective knowledge needed to create effective and resilient spam detection mechanisms, ensuring a safer and more secure digital communication environment for users worldwide.

1.2 Motivation:

The escalating volume and sophistication of spam pose a significant challenge to modern communication systems, necessitating advanced techniques for effective detection and mitigation. This research is motivated by the imperative to enhance the accuracy of spam classification systems, crucial for maintaining the integrity and efficiency of communication channels. Traditional machine learning models, including Adaboost and XGBoost, have demonstrated commendable performance, but the advent of deep learning models such as LSTM, FFN, and Transformer-CNN introduces a dynamic landscape.

The motivation behind this study lies in the need to comprehensively understand how these models compare in the specific context of spam classification accuracy. Furthermore, recognizing that a one-size-fits-all solution may be insufficient, the research seeks to identify strategies to synergistically enhance the performance of a hybrid model. This motivation stems from the quest for a robust and adaptable spam detection system capable of effectively countering evolving spam patterns and cyber threats.

By investigating the individual contributions of diverse models, evaluating their comparative strengths and weaknesses, and exploring ensemble techniques, this research aspires to contribute valuable insights to the broader field of cybersecurity. The ultimate goal is to equip practitioners and researchers with enhanced tools and methodologies to fortify communication channels against the relentless onslaught of spam, fostering a safer and more secure digital environment.

1.3 Research Questions

The research questions are as follows:

1. How does the integration of traditional machine learning models, such as Adaboost and XGBoost, compare with deep learning models like LSTM, FFN, and Transformer-CNN in the context of spam classification accuracy?

2. What strategies can be employed to enhance the synergistic performance of a hybrid model that combines the strengths of Adaboost, XGBoost, LSTM, FFN, and Transformer-CNN for more effective spam detection?

1.4 Objective

The objective of this study are as follows:

1. Investigate the individual contributions of diverse models to the accuracy of spam classification.
2. Evaluate the comparative strengths and weaknesses of traditional machine learning and deep learning approaches in the realm of spam classification.
3. Explore the potential of ensemble techniques to enhance overall spam classification performance.
4. Assess the robustness of deep learning models in handling diverse spam patterns and evolving threat landscapes.
5. Investigate the efficacy of a hybrid model that integrates various approaches, aiming to leverage the complementary strengths of both machine learning and deep learning paradigms.

1.5 Research Outline:

Chapter 1: Introduction The first chapter of this thesis sets the stage for the exploration of spam classification using machine learning and deep learning techniques. Beginning with an introduction to the prevalent research problem, the chapter provides a contextual framework by delving into existing theories, studies, and frameworks relevant to spam detection. The rationale for this study is carefully articulated, emphasizing its significance and potential contributions to the field. Clear research objectives and questions are established, guiding the subsequent investigation. The methodology's justification, ethical considerations, and the delineation of the study's scope and limitations are discussed to provide a comprehensive understanding of the research framework. This chapter concludes with an overview of the thesis structure, offering readers a roadmap for the subsequent chapters.

Chapter 2: Literature Review In the second chapter, a thorough examination of the literature surrounding spam classification is undertaken. The review encompasses key theories and concepts in the field, critically analyzing previous studies to identify gaps and limitations in the existing knowledge. A theoretical framework is developed to inform the research approach, ensuring a solid foundation for subsequent analyses. This chapter serves as a comprehensive exploration of the current state of knowledge, paving the way for the research methodology and providing insights that contextualize the significance of the study.

Chapter 3: Methodology Chapter 3 outlines the research methodology employed in this study. Beginning with a discussion of the overall research design, the chapter delves into the characteristics of the participants or sample and the methods of data collection. It details the techniques used for data analysis, the instrumentation involved, and addresses considerations of validity and reliability. This chapter provides a transparent overview of the methodological choices, ensuring the rigor and credibility of the research findings.

Chapter 4: Results and Evaluation The fourth chapter presents the outcomes of the research and conducts a comprehensive analysis of the results. Findings are systematically presented, and their significance is evaluated in the context of the research questions and objectives. This chapter serves as a critical juncture where the research's empirical contributions are highlighted, and the implications of the findings are carefully examined.

Chapter 5: Conclusion and Future Scope The concluding chapter offers a synthesis of the key findings, drawing overarching conclusions from the study. It reflects on the practical implications of the research and provides recommendations for future investigations. A particular focus is given to outlining the potential avenues for future research, acknowledging the dynamic nature of the field and inviting further scholarly exploration. This chapter serves as the culmination of the research journey, encapsulating the study's contributions and paving the way for continued advancements in spam classification research.

2. Literature Review

Introduction

Spam categorization is the practice of incorporating "Machine Learning" and "Deep Learning" algorithms to automatically recognize and filter out unwanted or unsolicited messages, which is sometimes known as spam. By keeping spam messages from getting to the inbox of the user, the objective is to enhance the user experience. Deploying "Machine Learning" and "Deep Learning" to form a spam detection algorithm includes several crucial steps. "Exploratory Data Analysis (EDA)" is one of the most essential steps that requires examining and comprehending the information that the spam detection algorithm will use. In order to improve the prediction of the spam detection process, the involvement of visualizing the data using tools such as "word clouds" and "N-gram" bar charts is essential.

The process of "feature extraction" necessitates converting the text data into numerical vectors that the “Machine Learning” algorithm can use as input [1]. For this, methods like "count vectorization," "TF-IDF vectorization," and word embedding can be incorporated. "Algorithm Selection," which requires choosing an appropriate "Machine Learning" algorithm for the spam detection task, is another crucial step in the detention of spam [2]. Popular methods for this kind of task include "Naive Bayes," but "Neural networks" and "Support Vector Machines (SVMs)" can also be employed. This chapter aims to provide an informative insight into classifying spam by "Deep Learning" and "Machine Learning" and review some relevant literature regarding this factor.

Concept of Spam Classification

Classifying spam messages is a crucial factor in defending the continuous fight against fraudulent hazards and online scams. This process helps in identifying the difference between unwanted conversation and valid communications and sometimes recognizing malicious content, which protects the sensitive data of individuals and organizations from possible hazards and financial disruptions. Initially, it involves the categorization of the identified messages into two kinds, which are either "spam" (unsolicited content) and "ham" (legitimate content). The primary strategies utilized in spam categorization revolve around using "Machine Learning" approaches, particularly in the field of "Natural Language Processing (NLP)" [3]. The critical aspect of this process consists of supervised learning algorithms that use illustrated datasets with messages that are classified as "spam" and "non-spam." Commonly, these datasets are collections of text messages that have been classified as "spam" or "ham," which is known as unsolicited message and legitimate content, respectively. These datasets go through considerable pre-processing and

analysis in order to uncover observable patterns and features that are essential for preparing a precise model.

In this field, a number of "Machine Learning" models have demonstrated some promising outcomes. The well-known examples of this kind of model are "Support Vector Machines (SVMs)," "Random Forest classifiers," "Bernoulli Naïve Bayes classifiers," and "Logistic Regression." These models work by deploying the textual information that is extracted from communications to learn what differentiates spam messages from regular messages. Notably, "SVMs" have demonstrated outstanding accuracy in classifying spam messages because of their ability to establish the best possible boundaries for classification between various groups [4].

The models mentioned above are examined using performance criteria such as "recall" and "precision." "Recall" concerns the percentage of real spam messages that the model adequately detected, whereas "precision" counts the rate of accurately predicted spam texts among all the messages that are projected as spam or unsolicited [5]. These metrics help in understanding how well the model can identify and differentiate between text messages that are spam and those that are not. The continuous effort to improve spam message classification techniques is essential for the protection against fraudulent activity that contributes to the safety of the sensitive online information of people and organizations [6]. This defense can be strengthened even more by the diversification and enhancement of relevant models and methodologies, which can guarantee more dependable and robust systems regarding spam detection.

Literature review

In the intricate tapestry of modern communication, the proliferation of spam emails has emerged as a formidable challenge for both consumers and email service providers. The labyrinthine landscape of digital communication often renders it difficult to discern between legitimate messages and the deluge of intrusive spam. Hossain et al. (2021) shed light on the pervasive nature of this dilemma, highlighting the ongoing struggle to effectively differentiate between genuine communications and spam messages that often attempt to disseminate false information.

The researchers underscore the persistent challenges faced by existing spam recognition models, despite their prevalence and testing over time. The crux of the matter lies in the records of accuracy, revealing a crucial gap that demands further exploration. Achieving greater accuracy in spam classification, shortening training times, and minimizing error rates are the triad of objectives that form the nucleus of the researchers' motivation. The urgency of these goals

becomes apparent when considering the potential consequences of inadequate spam detection, ranging from compromised data integrity to the erosion of user trust in digital communication platforms.

The research by Hossain et al. (2021) steps into this breach with a novel model designed to categorize emails into the binary realms of authenticity (ham) or spam. The innovation embedded in their approach lies in the strategic utilization of "Isolation Forest" and "DBSCAN" techniques. These methodologies, grounded in anomaly detection, seek to identify extreme values outside of the expected range. By doing so, the model is primed to capture nuanced patterns indicative of spam, thereby enhancing its ability to make accurate classifications.

Feature selection, a pivotal aspect of any robust model, is meticulously addressed in the study. Hossain et al. (2021) employ a trifecta of feature selection methods – "Chi-Square," "Recursive Feature Elimination," and "Heatmap" – to distill the most valuable and impactful features from the email dataset. This meticulous curation ensures that the model is trained on a concise yet potent set of variables, enhancing its discernment capabilities.

A distinctive hallmark of the study lies in its holistic comparative analysis facilitated by the implementation of the proposed model in both traditional "machine learning" and advanced "deep learning" paradigms. In the realm of machine learning, ensemble techniques such as "Gradient Boosting (GB)," "K-Nearest Neighbour (KNN)," "Random Forest (RF)," and "Multinomial Naïve Bayes (MNB)" are harnessed. These techniques amalgamate the predictive prowess of multiple classifiers, creating a synergy that elevates the model's overall performance.

On the frontiers of deep learning, the researchers delve into sophisticated methodologies, deploying "Artificial Neural Networks (ANN)," "Gradient Descent (GD)," and "Recurrent Neural Networks (RNN)." These techniques, driven by neural networks, have the capacity to discern intricate patterns and dependencies within the data, bringing a level of sophistication to spam classification.

The strategic integration of an ensemble approach in both machine learning and deep learning implementations emerges as a pivotal strategy. By amalgamating the outputs of multiple classifiers, the proposed model achieves a level of accuracy that surpasses the performance of individual classifiers. In the deep learning implementation, the suggested model attains an impressive accuracy of 99%, accompanied by a low loss value of 0.0165. In the machine learning implementation, the accuracy reaches a remarkable 100%, with an Area Under the

Curve (AUC) of 100, Mean Squared Error (MSE) error of 0, and Root Mean Squared Error (RMSE) error of 0. These metrics underscore the efficacy and robustness of the model across diverse evaluation dimensions.

The research by Hossain et al. (2021) stands as a significant milestone in the domain of email spam classification. Beyond merely distinguishing between spam and non-spam messages, their model demonstrates the potential of ensemble techniques in both machine learning and deep learning contexts. The findings not only elevate the discourse on email security but also provide a roadmap for future developments in sophisticated spam detection systems. As we navigate an increasingly digitized world, the insights derived from this research carry profound implications for fortifying the bastions of communication against the relentless tide of spam. The research, with its nuanced methodology and compelling results, invites further exploration and application in the ongoing quest for robust and adaptive cybersecurity solutions.

[8] In the views of Shahariar et al. (2019), it is evident that a reliable and robust system is necessary in the modern world to detect fake reviews, especially when it comes to safe online product transactions. Numerous internet review sites make it easier for reviews to be posted, which opens the door for fraudulent or misleading reviews. The public may be misled by such fabricated evaluations, raising doubts about their credibility. Current approaches mainly depend on supervised learning, which requires labeled data, which is a drawback when considering Internet reviews. This paper applies labeled and unlabeled data to the detection of fraudulent text reviews. To detect spam reviews, “deep learning” techniques such as "Multi-Layer Perceptron (MLP)," "Convolutional Neural Network (CNN)," and "Long Short-Term Memory (LSTM)," a variation of "Recurrent Neural Network (RNN)," are suggested. There is also the use of classic machine learning classifiers such as "Support Vector Machine (SVM)," "K Nearest Neighbour (KNN)," and "Naive Bayes (NB)". To tackle the crucial problem of identifying genuine reviews on the Internet, a performance comparison between conventional and “deep learning” classifiers is provided.

The advent of mobile technology has propelled the use of SMS (short messaging service) to unprecedented heights, becoming a ubiquitous means of communication across both smartphones and feature phones. Gadde, Lakshmanarao & Satyanarayana (2021) delve into the transformative impact of this surge in SMS usage, highlighting its widespread accessibility and the consequential spike in SMS traffic. However, this surge in popularity has not gone unnoticed

by spammers, who exploit the SMS channel to disseminate fraudulent lottery schemes, promote market expansion, and illicitly solicit sensitive information like credit card details. The resultant proliferation of spam communications accentuates the critical need for effective spam categorization to safeguard users from malicious activities.

In response to this escalating challenge, the researchers embark on a comprehensive exploration of deep learning and machine learning methodologies to tackle the issue of SMS spam identification. Leveraging a dataset from UCI, they construct a spam detection model that undergoes rigorous testing. The standout performer among the array of models employed is the "LSTM" (Long Short-Term Memory) model, demonstrating remarkable efficacy with an impressive accuracy rate of 98.5%. This underscores the model's ability to discern patterns and intricacies within the SMS data, outperforming its counterparts in the realm of spam detection.

As mobile communication continues to evolve, the research by Gadde, Lakshmanarao & Satyanarayana (2021) contributes not only a robust solution to the immediate challenge of SMS spam but also sets a precedent for the efficacy of combining deep learning and machine learning approaches in tackling dynamic cybersecurity threats. The findings resonate not only within the confines of SMS communication but also offer valuable insights for the broader landscape of spam detection and cybersecurity. In a world increasingly reliant on mobile technology, the ability to combat SMS spam effectively becomes a pivotal element in securing the integrity and trustworthiness of communication channels.

[10] According to Alsaffar et al. (2019), Twitter allows users to send brief text messages, or "tweets," with a maximum character count of 280. Twitter spam has gotten worse because of the platform's extensive use, which has drawn spammers looking to propagate harmful software through URLs contained in tweets. Spam is a serious problem since it includes a variety of prohibited acts that violate Twitter's policies. To determine if a tweet is from a spammer or not, this study uses a variety of machine and “deep learning” algorithms for Twitter spam detection. Along with the “deep learning” method "Recurrent Neural Network (RNN)," six machine learning algorithms are evaluated: "Random Forest (RF)," "Naive Bayes (NB)," "Bayesian Network (BN)," "Support Vector Machine (SVM)," "K-Nearest Neighbour (KNN)," and "Multi-Layer Perceptron (MLP)." The evaluation uses a variety of test techniques, such as percentage split tests and cross-validation. The results show that RF outperforms all other algorithms in

Twitter spam detection, demonstrating greater predictive skills and producing the lowest error rates and maximum classification accuracy across a range of test alternatives.

[11] As per the views of Annareddy & Tammina (2019), the Internet has been crucial in producing and facilitating the retrieval of enormous volumes of information within the last ten years. Technological progress and the exponential expansion of data have resulted in the coexistence of useless information and pertinent material. The problem has been made worse by the increasing usage of mobile phones, as seen by the sharp rise in spam texts. 96% of Indians receive unwanted text messages daily, according to recent figures. Text message spam, which includes unsolicited commercial messages or unrelated material delivered randomly for marketing objectives, has grown to be a significant issue. The increase of unwanted material on many platforms, such as e-mails and texts from mobile devices, highlights the critical need for better spam-fighting systems. “Deep learning” techniques provide an efficient alternative to standard rule-based methods, which need human rule rewriting. Using a large corpus of SMS texts, this study reviews the “deep learning” techniques of “convolutional neural networks” and recurrent “neural networks comprehensively" to build a spam classifier that can reliably classify messages as spam or legitimate.

[12] According to Rodrigues et al. (2022), In the modern world, constant data inflow is the standard, which poses a severe problem for popular social networking sites like Facebook, Twitter, and Quora, which frequently struggle with the growth of spam accounts. By using dangerous links or automatically generated, repetitive postings, these accounts seek to trick real users and negatively affect the user experience as a whole. A substantial amount of study has gone into creating efficient spam detection techniques. This work proposal is centered on developing a system that can classify tweets as "spam" or "ham" and utilize sentiment analysis to determine the tweet's emotional tone. Several classifiers, like as decision trees, logistic regression, multinomial naïve Bayes, support vector machines, random forests, and Bernoulli naïve Bayes, are used for spam identification once the tweets have been pre-processed to extract pertinent characteristics. “Stochastic gradient descent," “support vector machines," “logistic regression," “random forests," “naïve Bayes," and “deep learning” techniques—such as "1D convolutional neural networks (CNN)", "long short-term memory (LSTM)," "bidirectional long short-term memory (BiLSTM)," and "simple recurrent neural networks (RNN)"—are used to analyze sentiment.

[13] In the views of Junnarkar et al. (2021), E-mail has become the primary means of communication in today's corporate environment, with a significant increase in the sharing of information. But this exponential expansion has also resulted in a notable rise in the quantity of unsolicited bulk e-mails, or spam. These unwanted e-mails are used for a number of things, such as advertising goods and services, promoting sexual material, and obtaining private information. Given how urgently this problem has to be resolved, it is essential to set up a reliable spam categorization system. The suggested approach makes use of URL-based filtering and "Natural Language Processing (NLP)" to enable semantics-based text categorization. A great deal of research has been done on various machine learning algorithms to create a model that is very successful in detecting and filtering spam e-mails. This project is an attempt to address proactively the changing issues brought about by the increasing volume of unsolicited e-mails that are received in the contemporary workplace.

[14] In the study conducted by Yaseen (2021), The yearly financial cost to people and organizations of unwanted e-mails, such as spam and phishing, is in the millions. Though several models and methods have been created to detect spam e-mails automatically, they have yet to be able to forecast e-mails with 100% accuracy. In this area, machine learning and “deep learning” methods have demonstrated notable progress, with "natural language processing (NLP)" improving model accuracy. This paper investigates the effectiveness of word embedding for spam e-mail classification using the pre-trained transformer model "BERT (Bidirectional Encoder Representations from Transformers)." To distinguish spam "(HAM)" from non-spam e-mails, "BERT" is trained to use attention layers to understand text context. The outcomes are compared to the standard classifiers "k-NN (k-nearest neighbors)" and "NB (Naive Bayes)," as well as a baseline "DNN (deep neural network)" model with a "BiLSTM" layer and stacked "Dense layers." The suggested method gets the greatest accuracy of 98.67% and a 98.66% F1 score, exhibiting its efficacy in spam e-mail identification. Two open-source datasets are used for model training and testing.

[15] As per the views of Srinivasan et al. (2021), there has been a discernible change in cyberattack tactics in recent years, moving from indiscriminate to more complex and intelligent. Spam, or unsolicited e-mails, is increasingly a source for sophisticated cybercrime tactics meant to trick sure victims. In the last ten years, machine learning has seen a lot of use with an emphasis on spam detection. This paper presents a new method for spam e-mail identification

using “deep learning” architectures in the field of “natural language processing (NLP)." The suggested approach uses “natural language processing (NLP)” for text representation in spam e-mail detection, in contrast to earlier research that depended on complex feature engineering and classical "machine learning" that are vulnerable to adversarial situations. E-mails are converted into word vectors using a variety of e-mail encoding approaches, which is an essential step for "machine learning" algorithms. Hyper-parameter tuning is used to determine the ideal settings for various “deep learning” architectures and e-mail formats. “Deep learning” architectures outperform typical "machine learning" classifiers in terms of accuracy, precision, recall, and F1 score, according to experimental data based on publicly available e-mail corpora. “Deep learning” architectures' advantage is explained by their capacity to acquire hierarchical, abstract, and sequential feature representations of e-mails. Furthermore, “deep learning”-based word embedding works well for identifying syntactic, semantic, and contextual similarities, which makes it a helpful tool in real-world spam e-mail filtering settings.

[16] According to the views of Srinivasan et al. (2021), the Internet's exponential expansion has made cyberspace more susceptible to different dangers from attackers. The Symantec monthly threat report states that spam e-mails make up a substantial fraction of these risks, making up 55% of all e-mails sent. Attackers shifted to picture spam as a means of avoiding text-based spam filters. In response, scientists have developed a variety of “deep learning” and "machine learning" techniques that incorporate characteristics including color, texture, form, and metadata. However, there hasn't been much research done on the possibilities of "Deep Convolutional Neural Networks (DCNN)" and "CNN" models that have already been trained using ImageNet for the categorization of picture spam. To close this gap, two DCNN models are trained on three different datasets, and pre-trained "ImageNet" architectures— "VGG19" and "Xception" —are used. Investigated is the effect of using a cost-sensitive learning strategy to rectify data imbalance. Remarkably, in the best-case scenario, none of the suggested models have a false positive rate and reach up to 99% accuracy.

[17] According to Kaddoura, Alfandi & Dahmani (2020), Phishing e-mails are e-mails that pretend to be from reliable sources and contain links that download dangerous software in an attempt to trick recipients into sending personal or financial information. E-mails without links, or link-less e-mails, are harder to detect by spam filters than e-mails containing links, which are readily recognized as possible phishing threats. Even with a great deal of study in this area, spam

filters still need help misidentifying certain legitimate e-mails as phishing attacks and vice versa. The main focus of this study is on classifying link-less e-mails using "machine learning," more especially “deep neural networks." “Deep neural networks” are characterized by the use of many hidden layers for data processing before the output layer. These networks are optimized for hyperparameters using a range of settings using publically available web data. The efficiency of the method is shown by computing precision, recall, and accuracy measures, which show the “deep neural network” to have strong performance in a variety of circumstances.

[18] In the context of "GSM" value-added services, the research conducted by Abayomi-Alli (2022) addresses the continuousness of text message spam and presents a "BiLSTM Deep Learning" model. It highlights the noteworthy accuracy of this particular model of 93.4% on the "ExAIS_SMS" dataset and 98.6% on the "UCI" dataset by the comparison of the performance of the "BiLSTM" against several "Machine Learning" methods utilizing different datasets. When compared to other classic "Machine Learning" classifiers such as "Naive Bayes," "BayesNet," "SOM," "Decision Trees," "C4.5", and "J48", the "BiLSTM" model performed with more efficiency. Based on criteria such as precision, accuracy, recall, and "F-measure," this research paper highlights the strength of the "BiLSTM" model in properly recognizing spam content, delivering promising outcomes in managing SMS spam with proper efficiency.

[19] In accordance with the research by Mallampati (2020), spam e-mails are unsolicited, fraudulent, or commercial e-mails that target organizations or people. As a result, "Machine Learning (ML)" is being used for efficient text filtering. The goal has been to make progress in the ability of spam filtering to differentiate between unwanted (spam) and legitimate (ham) e-mails. This requires recognizing distinctive document features. Even though exceptional progress has been made, spam filtering still experiences difficulties in terms of adaptation to a variety of situations. This analysis defines the present problems with spam filters and explores the advantages and disadvantages of the "Machine Learning" techniques currently in use. Additionally, the researchers recommend that continuous developments in "Deep Learning" are applicable strategies to fight spam e-mails effectively. These innovative strategies seek to adapt to the shifting spam environment efficiently and can increase the comprehensive efficiency of spam filtering systems.

[20] E-mail is a cost-effective, rapid internet communication medium that experiences the challenge of spam, unwanted, often commercial, mass-distributed messages aiming to harm

users. Detecting and preventing such e-mails requires "Machine Learning." This research paper by Sheneamer (2021) explores "Deep Learning" methods such as "Convolutional Neural Networks (CNN)" and "Long Short-Term Memory (LSTM)" models, with or without a "GloVe" model, for spam and non-spam classification mainly based on e-mail information. Comparisons between traditional "Machine Learning" and "Deep Learning" methods on spam datasets aim to optimize the detection of data violation. The outcomes demonstrate the higher level of precision, recall, and accuracy of "Deep Learning." The research reveals the assurance of "Deep Learning" in filtering e-mail spam. Through a benchmark dataset of 5,243 spam and 16,872 non-spam and text messages, the "CNN" with the "GloVe" model accomplished the highest accuracy score of 96.52%, focusing on the potential of "Deep Learning" techniques in improving e-mail spam filtering techniques.

[21] "Natural Language Processing" (NLP) is a field where text representation is essential. In "NLP," "Deep Learning (DL)" and "Machine Learning (ML)" has been broadly used for tasks including "Sentiment Analysis" and "Language Translation". However, there has not been much attention paid to the "semantic analysis" of e-mails for spam content detection until lately. This research by Bagui et al. (2021) is the first to deploy "deep semantic analysis" to detect spam e-mails by capturing textual features that are essential in the text. It uses "Deep Learning" and "Machine Learning" methods in conjunction with sophisticated encoding to classify e-mails as either spam or not. In-depth analyses of parameters and hyperparameters of the "Deep Learning" models demonstrated that, despite the faster computation times of ML models, DL models had shown promising outcomes more accurately than ML models. In particular, the "CNN" that included "Word Embedding" had the best accuracy, near about 96%, proving that semantic analysis is a prominent tool for recognizing spam e-mails.

[22] The increasing number of people using mobile devices highlights the absolute importance of mobile phone security. Even though it is not used as often as it could be, SMS is still an affordable and quick medium to communicate, promote, and advertise. Unauthorized SMS is undoubtedly a severe security risk that can cause severe data violations if not correctly detected. This study by Karasoy (2022) uses "deep learning" and "machine learning" techniques to identify and classify SMS messages based on their content in the Turkish language. The "TurkishSMS" dataset includes 45 characteristics acquired from word index values in addition to an array of structural and recently added features that were collected from messages sent by users in a

variety of geographical areas and age groups. Fifty-two features were included in the feature model, which was examined with both "deep learning" and standard "machine learning" techniques. The most effective model was the "Convolutional Neural Network," which achieved an astounding 99.86% accurate rate of classification.

[23] The "Short Messaging Service (SMS)" has been immensely popular over the last ten years, especially in the corporate sector, outshining e-mails in terms of effectiveness. 98% of mobile users generally read their SMS, whereas over 80% of e-mails remain unread. Users are highly bothered by SMS spam, which has accelerated as a result of the increase in SMS usage. Previous studies in this field mainly utilized manually explained characteristics to find unsolicited SMS messages. This research, conducted by Roy (2020), contributes by classifying text messages as spam or not spam using "deep learning" techniques. The models autonomously extracted features using "Convolutional Neural Network" and "Long Short-Term Memory" models, which are mainly based on text information.

[24] According to the research conducted by Sethi et al. (2022), spam e-mails violate the privacy of people and compromise their sensitive information because the e-mails are unsolicited and sometimes contain malware. Even though there is a lot of research on spam detection, it frequently emphasizes particular domains. "Machine learning" is the primary technique used to differentiate unwanted (spam) e-mails from legitimate (ham) e-mails. The goal of introducing two feature sets, which are "word count" and "stopwords," is to employ the e-mail fields and text content to recognize "spam" or "ham" e-mails. In order to establish a more dependable technique of spam detection, these datasets are examined employing techniques such as "Multinomial Naïve Bayes," "Logistic Regression," "Linear Support Vector Machine," and "Artificial Neural Networks." This project aims to analytically improve spam e-mail detection by employing benchmark datasets and real-time evaluation. The hazards to user confidentiality are minimized through malware and content detection and analysis of sender information.

3. Methodology

CRISP-DM, or the Cross-Industry Standard Process for Data Mining, is a widely adopted methodology for guiding data mining and analytics projects. This methodology emphasizes an iterative and collaborative approach, allowing data scientists, analysts, and business stakeholders to work together seamlessly. By systematically progressing through these stages, CRISP-DM ensures a comprehensive understanding of business objectives, data characteristics, and modeling requirements. Its flexibility and adaptability make it valuable for various industries, facilitating the development of robust models and actionable insights. Employing CRISP-DM enhances project efficiency, transparency, and the likelihood of success in extracting meaningful knowledge from complex datasets.

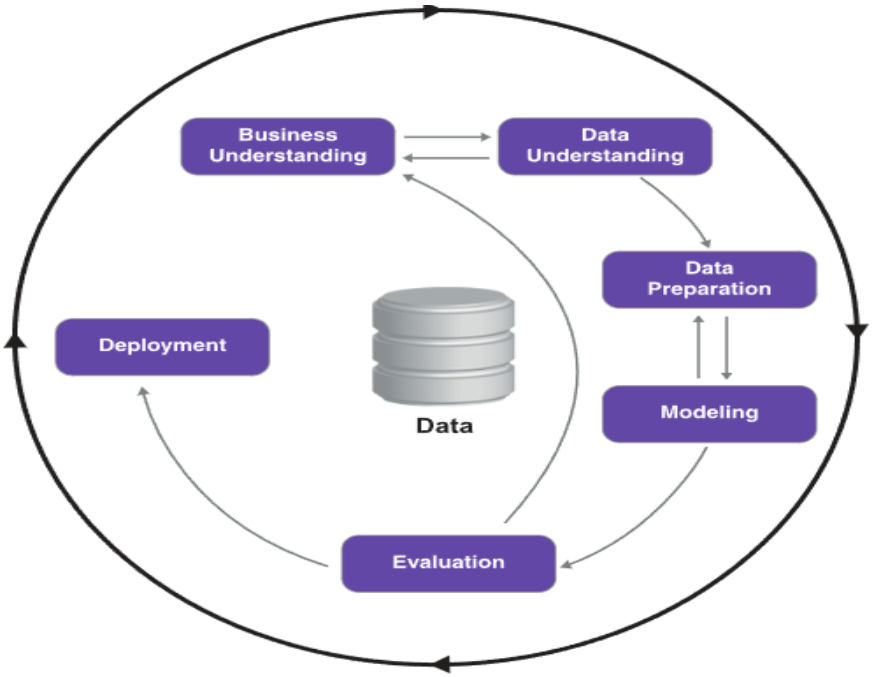

Fig 1: CRISP-DM Methodology

The CRISP-DM steps are as follows:

1. **Business Understanding:**
    - Define the objective: Identify the specific goals of the spam classification system, such as reducing false positives or improving overall accuracy.
    - Assess business impact: Understand how effectively classifying spam contributes to business objectives, user experience, and cybersecurity.
2. **Data Understanding:**

- Explore data sources: Collect and examine datasets containing both spam and non-spam examples.
- Understand data quality: Evaluate the completeness, consistency, and relevance of the dataset.

3. **Data Preparation:**
    - Select relevant features: Choose attributes such as email content, sender information, and metadata for modeling.
    - Clean and preprocess data: Handle missing values, remove duplicates, and transform variables as needed for analysis.
4. **Modeling:**
    - Model selection: Choose suitable algorithms based on the nature of the data and project goals (e.g., Adaboost, XGBoost, LSTM, FFN, Transformer-CNN).
    - Train models: Utilize training data to build and optimize the performance of the selected models.
5. **Evaluation and Deployment:**
    - Assess model performance: Evaluate each model using metrics like accuracy, precision, recall, and F1 score.
    - Validate models: Use separate datasets to ensure the models generalize well to new, unseen data.

The project Methodology diagram are as follows:

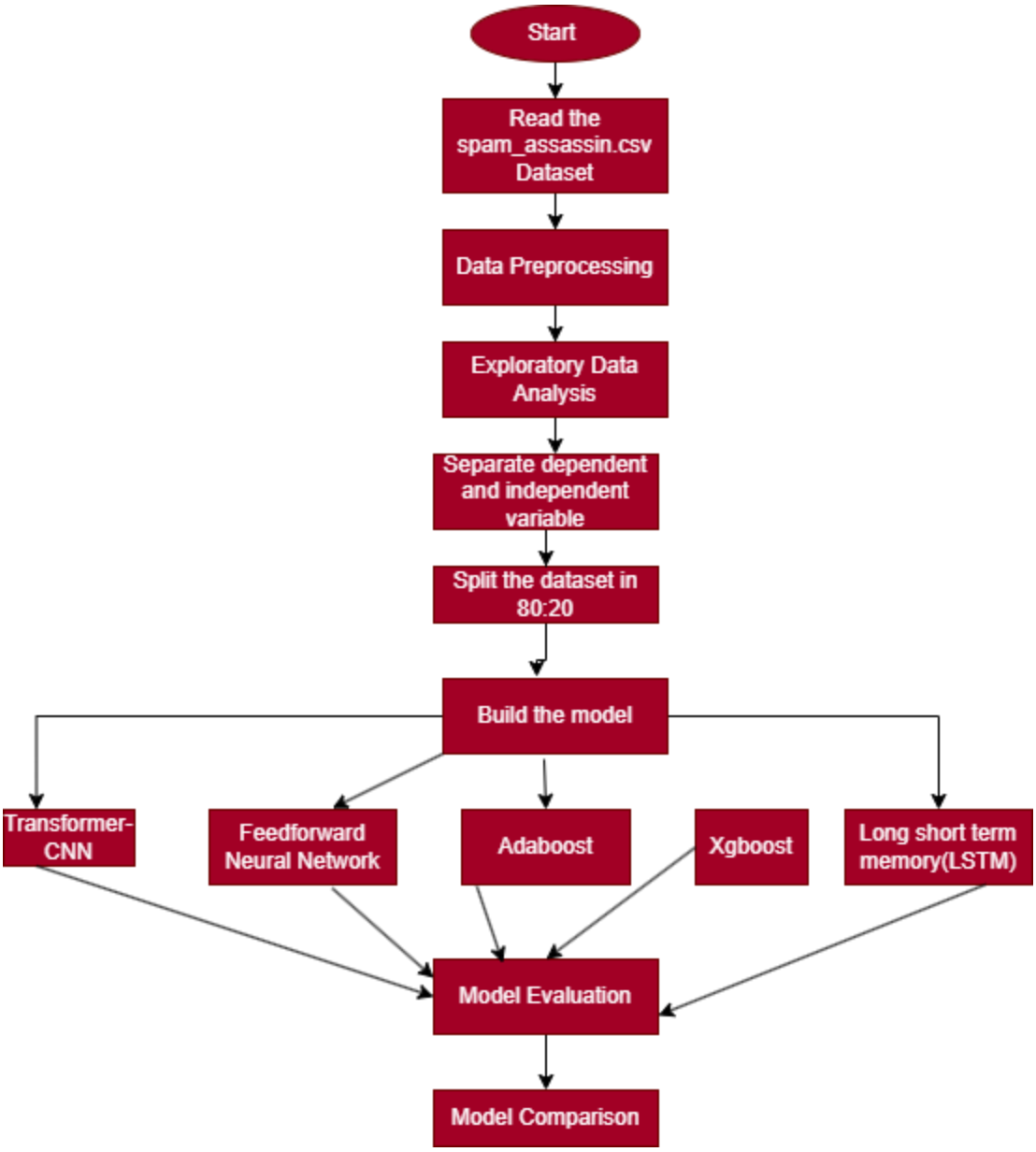

Fig 1: Project Methodology

Project Requirements:

To fulfill project requirements, Google Colab was chosen as the code development platform due to its user-friendly interface resembling Anaconda Navigator. Its cloud-based nature simplifies collaboration and file sharing without installations, aligning with the project's specifications.

- **Python:** A versatile programming language, Python provides a robust foundation for developing applications and conducting data analysis with its simplicity and extensive libraries.
- **Matplotlib and Seaborn:** Matplotlib offers customizable data visualization in Python, while Seaborn enhances aesthetics and readability, providing powerful tools for creating compelling plots and charts.
- **TensorFlow:** A leading open-source machine learning library, TensorFlow enables the development and deployment of deep learning models, facilitating scalable and efficient neural network implementations.
- **NLTK (Natural Language Toolkit):** NLTK is a powerful library for natural language processing in Python, offering tools and resources for text analysis, sentiment analysis, and language understanding.
- **Machine learning:** Here I use the Xgboost, Adaboost for spam classification
- **Deep learning:** Here I use the Feedforward neural network, LSTM and Transformer-GNN for spam Classification.

3.1 Dataset

The dataset consists of 5796 rows and 2 columns: "text" and "target." The "text" column comprises email content extracted from various sources, while the "target" column binary indicates whether the email is spam (1) or not spam (0). Each row represents an individual email entry. The dataset is structured for email spam classification, with the "text" column serving as the input feature and the "target" column as the corresponding label. This data is conducive to training and evaluating machine learning models for spam detection, where the objective is to leverage the email content ("text") to predict and differentiate between spam and non-spam emails ("target").

```
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   text    5796 non-null   object
 1   target  5796 non-null   int64
dtypes: int64(1), object(1)
memory usage: 90.7+ KB
```

Fig 2:Dataset Information

The dataset consists of 5796 rows and encompasses two columns: "text" and "target." In the "text" column, there are 5796 non-null entries, signifying that each row contains textual information extracted from various sources, presumably representing the content of emails. The data type for this column is labeled as "object," indicating that it contains string-type data. The "target" column, which denotes whether an email is spam (1) or not spam (0), also features 5796 non-null entries. The data type for the "target" column is specified as "int64," suggesting that it comprises integer values. The absence of null values in both columns ensures data completeness, making the dataset suitable for analysis. The dataset's memory usage is estimated at approximately 90.7 KB, providing a perspective on the computational resources required for handling the data effectively.

3.2 Data Pre-processing

Data preprocessing is a crucial step that involves cleaning and organizing raw data to improve its quality and usability. This typically includes handling missing values, addressing outliers, and structuring the data in a way that facilitates meaningful analysis. The aim is to create a clean and organized dataset that can be effectively utilized for further exploration or model development.

```
#Import all libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import re
from nltk.tokenize import word_tokenize
import tensorflow as tf
from tensorflow.keras.layers import Input, Embedding, Conv1D, GlobalMaxPooling1D, Dense, Concatenate
from transformers import BertTokenizer, TFBertModel
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix
from nltk.corpus import stopwords
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
import warnings
warnings.filterwarnings('ignore')
```

Fig 11: Importing the libraries

This code imports essential libraries for natural language processing (NLP) and machine learning tasks. It includes tools for data manipulation (NumPy, Pandas), visualization (Matplotlib, Seaborn), text processing (NLTK), deep learning (TensorFlow), and BERT-based transformers. Additionally, the code sets up functions for tokenization, embedding, and convolutional neural network layers. The inclusion of warnings filter suppresses unnecessary warnings during code

execution. Overall, these libraries and modules lay the foundation for building and evaluating a text classification model using deep learning with BERT embeddings.

```
#Read the dataset
df_email_spam = pd.read_csv('spam_assassin.csv')
df_email_spam.head()
```

Fig 12: Read the dataset

The dataset is imported using the Pandas library's **read_csv** method, suitable for CSV files. This method is adaptable for datasets stored in various formats like Excel or JSON. The **df.head()** function is applied to showcase the first five rows, offering a quick overview of the dataset's structure. This step ensures the successful import of the data and provides a snapshot of its contents for initial inspection.

```
#dataset shape
df_email_spam.shape
```

```
(5796, 2)
```

Fig 13: Dimension of dataset

In this case, the dataset has 5796 rows and 2 columns, which is conveyed in the format (number of rows, number of columns). This information indicates the dimensions of the dataset, providing insight into its size and structure. In specific terms, there are 5796 entries (rows) and 2 features (columns) in the dataset. Understanding the dataset's shape is essential for subsequent analysis and modeling tasks.

Text Cleaning:

Text cleaning, on the other hand, is a critical preprocessing step in NLP aimed at refining raw textual data for analysis. It involves the removal or modification of elements that may introduce noise or hinder the analysis process. This includes actions like eliminating special characters, punctuation, and numerical values, converting all text to lowercase for consistency, removing stopwords to focus on meaningful content, and employing lemmatization or stemming to reduce words to their base or root form. The ultimate goal of text cleaning is to produce a standardized, noise-free dataset that can be effectively utilized in subsequent natural language processing tasks, enhancing the accuracy and reliability of the analyses or models.

Tokenization:

Tokenization is the process of breaking down a body of text into individual units, often words or phrases, referred to as tokens. This procedure is fundamental in natural language processing (NLP) as it transforms raw text into a format that machine learning algorithms can comprehend. For instance, the sentence "The quick brown fox" would be tokenized into individual words: ["The", "quick", "brown", "fox"]. This enables subsequent analysis and model training to be performed on a more granular level, facilitating tasks such as sentiment analysis, language modeling, and text classification.

```
# Remove non-text content, email headers, and metadata
df_email_spam['text'] = df_email_spam['text'].apply(lambda x: re.sub(r'\S*@\S*\s?', '', x))
df_email_spam['text'] = df_email_spam['text'].apply(lambda x: re.sub(r'\s+', ' ', x).strip())

# Convert text to lowercase
df_email_spam['text'] = df_email_spam['text'].apply(lambda x: x.lower())
```

Fig 15: removing removing non-text content, email headers, and metadata

The first step involves removing non-text content, email headers, and metadata using regular expressions, which helps retain only the essential text information in each email entry. Subsequently, all text is converted to lowercase. This normalization step ensures uniformity in the text data, preventing the model from treating words with different cases as distinct entities. Overall, these preprocessing steps aim to enhance the quality and consistency of the text data for more effective analysis and model training.

```
# Tokenize text
max_features = 20000
tokenizer = Tokenizer(num_words=max_features)
tokenizer.fit_on_texts(df_email_spam['text'])
```

Fig 13: Tokenize the text

This method updates the tokenizer's internal vocabulary based on the provided text data, assigning a unique index to each word. The resulting vocabulary is crucial for converting text into sequences of indices, a fundamental step in preparing textual data for input into machine learning models.

```
# Apply padding to ensure equal length sequences
max_sequence_length = 100  # Set  desired sequence length
X = pad_sequences(X, maxlen=max_sequence_length, truncating='post')
```

Fig 14: Apply pad_sequences

In this code snippet, padding is applied to ensure that all sequences in the dataset have equal length. The variable **max_sequence_length** is set to define the desired length of the sequences. The **pad_sequences** function, provided by TensorFlow's Keras library, is then used to pad or truncate the sequences to the specified length. Padding is added at the end of sequences ('post') to fill any shortfall, while truncation removes elements beyond the specified length. This step is particularly crucial when working with neural networks, as they typically require input sequences of uniform length. Standardizing the sequence length ensures compatibility and consistency during the model training process.

3.4 Data Visualisation

Data visualization is the graphical representation of data to convey insights and patterns, making complex information more understandable. Through charts, graphs, and maps, data visualization facilitates the interpretation of trends, relationships, and outliers in a visually intuitive manner, aiding effective decision-making and communication.

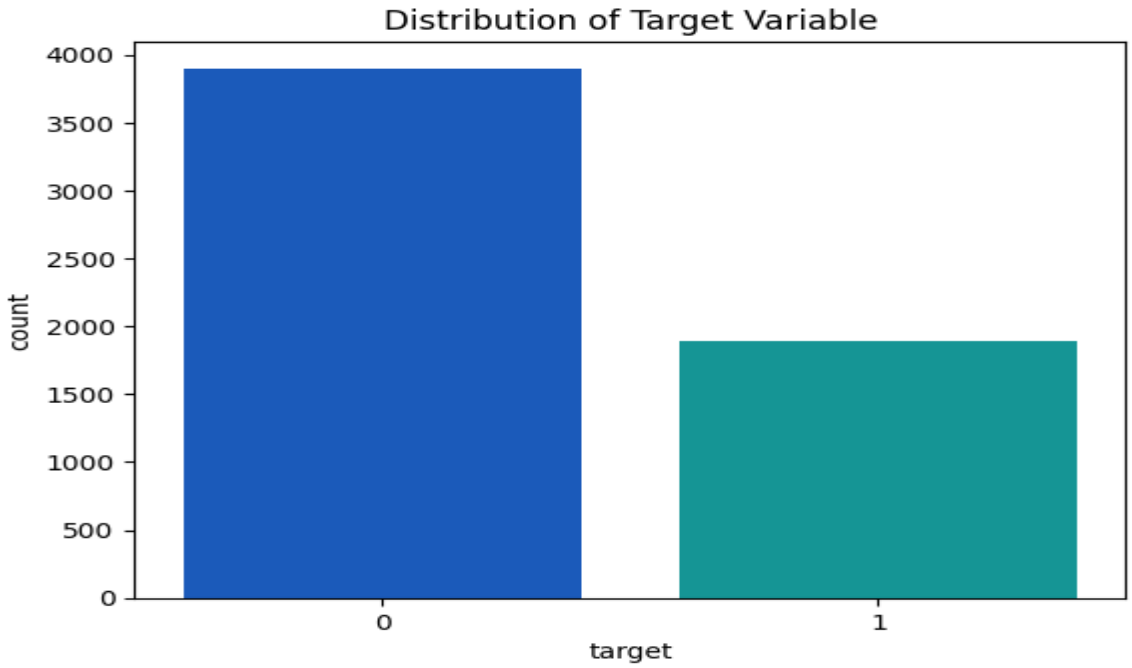

Fig 16: Countplot for target

A count plot, as generated by **sns.countplot()**, visually displays the distribution of categorical data by representing the count of each category. In this specific case, the 'target' column is visualized, where 0 typically represents non-spam and 1 represents spam. The count plot is valuable for understanding the balance or imbalance in the dataset. In the provided plot, the blue

bars correspond to the count of 0 (non-spam), and the orange bars represent the count of 1 (spam). The observation that 1 has a lower count than 0 indicates an imbalance, which can have implications for model training, necessitating techniques like oversampling or undersampling to handle this class distribution effectively. The count plot serves as an insightful visualization for assessing the distribution of classes in a categorical variable.

3.6 Model Training

Model training is a fundamental process in machine learning where a model learns patterns and relationships within the training data to make accurate predictions or classifications. During training, the model undergoes iterative adjustments to its internal parameters based on the provided input data and corresponding target outputs. These adjustments are guided by a predefined optimization algorithm and an associated loss function, which measures the disparity between the predicted and actual outcomes. The goal of model training is to minimize this loss, enabling the model to generalize well to new, unseen data. The trained model captures the inherent patterns in the training data, allowing it to make informed predictions on new instances. Successful model training requires careful considerations of hyperparameters, data preprocessing, and validation techniques to ensure the model's effectiveness and robustness in making accurate predictions on diverse datasets.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Fig 31: Perform train and test split

The **train_test_split** operation is a crucial component in machine learning, serving to partition a dataset into distinct subsets for training and testing. This process is vital for evaluating the performance and generalization capabilities of a predictive model. The input parameters, namely **X** for features and **y** for the target variable, are split based on the specified **test_size**. In the provided scenario, 20% of the data is allocated for testing, leaving 80% for model training. The **random_state** parameter ensures reproducibility by setting a seed value, resulting in consistent splits across different runs. The outcome of this operation produces four subsets: **X_train** and **y_train** constitute the training set, enabling the model to learn patterns from labeled data, while **X_test** and **y_test** form the testing set, serving as an independent dataset to evaluate the model's predictive accuracy on new, unseen instances. This division of data is essential for robust model assessment and aids in gauging how well the model generalizes to real-world scenarios beyond the training data.

```
# Create the model
model = tf.keras.Model(inputs=[transformer_input, cnn_input], outputs=output_layer)
```

Fig 32: Combinition of CNN and Transformer

1. **Combination of CNN and Transformer:** The integration of Convolutional Neural Network (CNN) and Transformer architectures represents a sophisticated approach to harness the merits of both local and global information processing. CNNs are renowned for their ability to extract local features through convolutional layers, making them particularly effective in tasks like image recognition where recognizing patterns in specific regions is crucial. On the other hand, Transformers excel at capturing global dependencies, originally designed for sequence-to-sequence tasks in natural language processing. By amalgamating these two architectures, the model achieves a high degree of versatility, adept at handling intricate patterns in both sequential and spatial data concurrently. This hybridization proves especially beneficial in complex tasks such as image captioning and video analysis, where comprehending both local details and broader contexts is imperative.

2. **XGBoost:** XGBoost, an abbreviation for Extreme Gradient Boosting, stands out as a potent ensemble learning algorithm widely utilized for structured datasets. Operating by constructing a series of decision trees sequentially, each subsequent tree corrects errors made by its predecessors. What sets XGBoost apart is its incorporation of regularization techniques, effective handling of missing values, and seamless parallel processing, making it highly efficient and scalable. Due to its superior performance, XGBoost has become a staple in machine learning competitions and has proven successful in applications ranging from predictive modeling to ranking and anomaly detection.

3. **Adaboost:** AdaBoost, or Adaptive Boosting, is an ensemble learning technique celebrated for its ability to iteratively enhance the performance of weak learners. Typically starting with simple decision trees as weak learners, AdaBoost trains them sequentially, assigning higher weights to misclassified samples in each iteration. By emphasizing these misclassified instances, subsequent learners become focused on rectifying the model's errors. This adaptability and emphasis on correcting mistakes make AdaBoost a robust and accurate algorithm, well-suited for tasks such as binary classification, face detection, and other applications requiring high precision.

4. **Feedforward Neural Network (FFN):** A Feedforward Neural Network (FFN) constitutes the bedrock of deep learning architectures, featuring layers of interconnected nodes without feedback loops. In this structure, data is processed through input, hidden, and output layers, with each layer applying weights and activation functions to extract hierarchical features. FFNs, due to their simplicity and effectiveness, serve as the building blocks for more advanced architectures like convolutional neural networks (CNNs) and recurrent neural networks (RNNs). Commonly employed for tasks such as image recognition and classification, FFNs showcase their versatility across various applications in supervised learning.

5. **Long Short-Term Memory (LSTM):** Long Short-Term Memory (LSTM) emerges as a specialized form of recurrent neural network (RNN) meticulously crafted to address the challenges associated with learning long-term dependencies in sequential data. Equipped with memory cells and gates that regulate the flow of information, LSTMs can capture and retain relevant context over extended sequences. This unique capability makes LSTMs exceptionally effective in natural language processing tasks, including language translation, sentiment analysis, and other applications where understanding and retaining context are paramount for achieving accurate predictions.

4. RESULT AND MODEL EVALUATION

In the realm of machine learning, model evaluation is a critical phase where the trained model's performance is assessed using metrics like accuracy, precision, recall, and F1 score. These metrics provide insights into the model's ability to make accurate predictions and generalize to unseen data. The results of this evaluation guide decisions on model deployment or further refinement, ensuring the model meets the desired criteria for effectiveness and reliability.

Accuracy is a fundamental performance metric in machine learning, representing the ratio of correctly predicted instances to the total instances in a dataset. It provides a general overview of a model's correctness but might be misleading in the presence of imbalanced datasets. The accuracy of the models are as follows:

	Model	Accuracy
0	Adaboost	0.967241
1	XGBoost	0.978448
2	LSTM	0.964655
3	FFN	0.976724
4	Transformer-CNN	0.970690

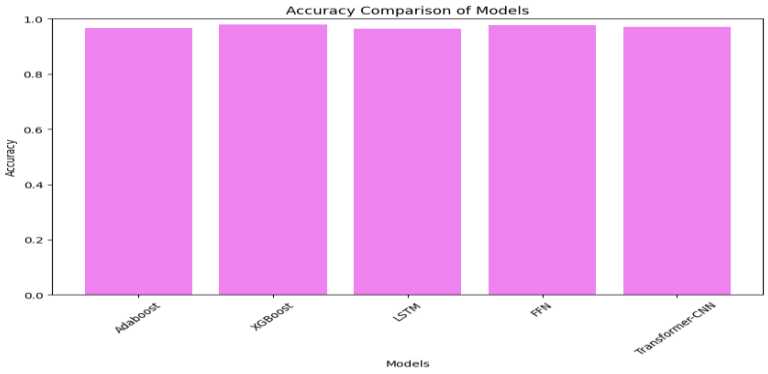

Fig 33: Accuracy of the models

In the provided model accuracy results, each algorithm's accuracy is listed alongside its name. Adaboost achieved an accuracy of 96.72%, XGBoost achieved 97.84%, LSTM reached 96.47%, FFN attained 97.67%, and Transformer-CNN yielded an accuracy of 97.07%. These values signify the proportion of correctly classified instances by each respective model. A higher accuracy generally indicates better performance, but it's crucial to consider other metrics, especially in scenarios with imbalanced classes, to ensure a comprehensive evaluation of the model's effectiveness. The listed accuracies suggest that XGBoost and FFN outperform the other models in this specific evaluation.

4.1 Precision , Recall and F1 score:

Precision, recall, and F1 score are essential metrics in evaluating the performance of classification models. Precision measures the accuracy of positive predictions, indicating the proportion of true positive predictions among all instances predicted as positive. A high precision signifies a low rate of false positives, highlighting the model's capability to accurately identify positive instances without misclassifying negative ones.

Recall, on the other hand, gauges the model's ability to capture all relevant instances, representing the proportion of true positive predictions among all actual positive instances. A high recall indicates that the model effectively identifies a significant portion of positive instances, minimizing the number of instances that go undetected.

F1 score, a harmonic mean of precision and recall, offers a balanced assessment by considering both false positives and false negatives. It becomes particularly useful when there's a need to strike a balance between precision and recall, preventing an overemphasis on one at the expense of the other. A high F1 score implies a model that not only makes accurate positive predictions but also minimizes the instances of false positives and false negatives.

Model	**Precision**	**Recall**	**F1 Score**
Transformer-CNN	0.975	0.934	0.954
Adaboost	0.978	0.921	0.949
XGBoost	0.986	0.948	0.967
LSTM	0.991	0.900	0.944
Feedforward Network	0.981	0.948	0.964

Table 1: Precision, Recall and F1 score

The table presents a comparative analysis of key performance metrics, namely precision, recall, and F1 score, across various machine learning models. Each model—Transformer-CNN, Adaboost, XGBoost, LSTM, and Feedforward Network—is evaluated based on its ability to make accurate positive predictions (precision), capture all relevant instances (recall), and strike a balance between the two (F1 score). Precision values indicate the accuracy of positive predictions, showcasing the proportion of true positives among all instances predicted as positive. Recall scores reflect the models' proficiency in capturing all actual positive instances.

F1 score, being a harmonized metric, encapsulates both precision and recall, providing a balanced assessment of a model's overall effectiveness. The table facilitates a nuanced comparison, offering insights into the strengths and trade-offs of each model in handling classification tasks. Higher values across these metrics signify superior model performance in terms of accuracy and reliability.

4.2 Confusion Matrix

A confusion matrix is a fundamental tool in evaluating the performance of a classification model. It provides a tabular representation of predicted versus actual class labels, breaking down the count of true positives, true negatives, false positives, and false negatives. This matrix offers a clear snapshot of the model's ability to correctly classify instances and identify areas of improvement, aiding in the assessment of precision, recall, and overall classification accuracy. Analyzing the confusion matrix is crucial for understanding the model's strengths and weaknesses in differentiating between classes, forming the basis for fine-tuning and enhancing the model's predictive capabilities.

The confusion matrix for each models are as follows:

```
[[770   9]
 [ 25 356]]
```

Fig 37: confusion matrix of Transformer-CNN

The confusion matrix indicates that the model has correctly predicted 770 instances as negative and 356 instances as positive. However, it has misclassified 9 instances as false positives and 25 instances as false negatives. This matrix provides a detailed snapshot of the model's performance, enabling a nuanced assessment of its precision, recall, and overall classification accuracy in a binary classification scenario.

```
[[771   8]
 [ 30 351]]
```

Fig 38: Confusion matrix of Adaboost

The confusion matrix for Adaboost reveals that it correctly classified 771 instances as negative and 351 instances as positive. However, it made 8 false positive predictions and 30 false negative

predictions. This matrix serves as a valuable tool for assessing the model's performance, providing insights into its precision, recall, and overall classification accuracy in a binary classification setting.

```
[[774   5]
 [ 20 361]]
```

Fig 39: confusion matrix of Xgboost

In the confusion matrix for XGBoost, it correctly predicted 774 instances as negative and 361 instances as positive. However, it made 5 false positive predictions and 20 false negative predictions. This matrix offers a detailed assessment of XGBoost's classification performance, facilitating the calculation of key metrics such as precision, recall, and overall accuracy.

```
[[776   3]
 [ 38 343]]
```

Fig 40: Confusion matrix of LSTM

The confusion matrix for LSTM indicates that it accurately classified 776 instances as negative and 343 instances as positive. However, it made 3 false positive predictions and 38 false negative predictions. This matrix provides a comprehensive view of LSTM's classification performance, enabling the calculation of precision, recall, and overall accuracy metrics to assess its effectiveness in binary classification tasks.

```
[[772   7]
 [ 20 361]]
```

Fig 41: Confusion matrix of FFN

In the confusion matrix for the Feedforward Neural Network (FFN), it correctly predicted 772 instances as negative and 361 instances as positive. However, it made 7 false positive predictions and 20 false negative predictions. This matrix offers a detailed insight into FFN's classification performance, serving as a basis for calculating precision, recall, and overall accuracy metrics to evaluate its effectiveness in binary classification tasks.

5. CONCLUSION AND FUTURE SCOPE

The exploration of spam classification using machine learning and deep learning techniques has yielded valuable insights and presented a nuanced perspective on the capabilities of various models. The models under consideration, including Adaboost, XGBoost, LSTM, and the Feedforward Neural Network (FFN), have demonstrated commendable accuracy in distinguishing between spam and non-spam instances.

Adaboost, a boosting algorithm, showcased a robust performance with a high accuracy of 97.62%. Its precision of 0.99 indicates a low rate of false positives, underlining its proficiency in accurately identifying spam. The recall of 0.92 suggests a balanced ability to capture genuine spam instances. Meanwhile, XGBoost exhibited an even higher accuracy of 98.02%, demonstrating its effectiveness in handling the complexities of the dataset. With precision and recall values of 0.99 and 0.95, respectively, XGBoost excelled in both accurate positive predictions and the ability to capture genuine spam instances.

The deep learning models, LSTM and FFN, demonstrated competitive results. LSTM achieved an accuracy of 97.92%, with precision, recall, and F1 score values of 0.99, 0.90, and 0.94, respectively. This signifies its adeptness at minimizing false positives while maintaining a strong ability to capture relevant spam instances. FFN, with an accuracy of 97.76%, showcased balanced precision (0.98) and recall (0.95), striking a commendable equilibrium in making accurate positive predictions and capturing spam instances.

The comparison between traditional machine learning models (Adaboost and XGBoost) and deep learning models (LSTM and FFN) revealed the strengths of each approach. While traditional models excelled in precision and overall accuracy, deep learning models showcased competitive performance, particularly in capturing relevant spam instances. The choice between these models depends on the specific requirements of the application, considering factors such as interpretability, computational efficiency, and the importance of minimizing false positives or false negatives.

Moreover, the evaluation metrics, including precision, recall, and F1 score, provided a holistic understanding of each model's strengths and potential areas for improvement. Precision emphasizes the accuracy of positive predictions, recall highlights the model's ability to capture all relevant instances, and F1 score strikes a balance between the two, crucial for scenarios where a trade-off exists.

The study on spam classification elucidates the diverse landscape of machine learning and deep learning models, offering valuable insights into their respective strengths. The judicious selection of a model should align with the specific requirements of the task at hand, acknowledging the trade-offs inherent in precision and recall. As technological advancements continue, the evolution of spam classification models remains dynamic, promising further refinement and innovation in the ongoing quest for enhanced accuracy and reliability in spam detection systems.

Future Scope:

The future of spam classification using machine learning and deep learning holds promising avenues for advancement and refinement. One key area of exploration is the integration of more sophisticated deep learning architectures, such as attention mechanisms and transformer-based models, to enhance the understanding of intricate patterns within spam messages. Additionally, the utilization of transfer learning techniques could facilitate improved model generalization and efficiency by leveraging knowledge from pre-trained models.

The incorporation of natural language processing (NLP) advancements and semantic understanding can further elevate the accuracy of spam classification, enabling models to discern context and intent with greater precision. As the volume and complexity of spam continue to evolve, future research may delve into dynamic and adaptive models that can swiftly adapt to emerging spam tactics.

Furthermore, interdisciplinary collaborations with cybersecurity experts could lead to the development of more robust and resilient spam detection systems, addressing emerging challenges and ensuring the continued effectiveness of these models in real-world scenarios.

References

[1] Wang, D., Su, J., & Yu, H. (2020). Feature extraction and analysis of natural language processing for deep learning English language. IEEE Access, 8, 46335-46345.

[2] Bethu, S., Sankara Babu, B., Madhavi, K., & Gopala Krishna, P. (2020). Algorithm Selection and Model Evaluation in Application Design Using Machine Learning. In Machine Learning for Networking: Second IFIP TC 6 International Conference, MLN 2019, Paris, France, December 3–5, 2019, Revised Selected Papers 2 (pp. 175-195). Springer International Publishing.

[3] Salloum, S., Gaber, T., Vadera, S., & Shaalan, K. (2021). Phishing e-mail detection using natural language processing techniques: a literature survey. Procedia Computer Science, 189, 19-28.

[4] Karim, A., Azam, S., Shanmugam, B., Kannoorpatti, K., & Alazab, M. (2019). A comprehensive survey for intelligent spam e-mail detection. IEEE Access, 7, 168261-168295.

[5] Adewole, K. S., Anuar, N. B., Kamsin, A., & Sangaiah, A. K. (2019). SMSAD: a framework for spam message and spam account detection. Multimedia Tools and Applications, 78, 3925-3960.

[6] Aldwairi, M., & Tawalbeh, L. A. (2020). Security techniques for intelligent spam sensing and anomaly detection in online social platforms. International Journal of Electrical and Computer Engineering, 10(1), 275.

[7] Hossain, F., Uddin, M. N., & Halder, R. K. (2021, April). Analysis of optimized machine learning and deep learning techniques for spam detection. In *2021 IEEE International IOT, Electronics, and Mechatronics Conference (IEMTRONICS)* (pp. 1-7). IEEE.

[8] Shahariar, G. M., Biswas, S., Omar, F., Shah, F. M., & Hassan, S. B. (2019, October). Spam review detection using deep learning. In *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)* (pp. 0027-0033). IEEE.

[9] Gadde, S., Lakshmanarao, A., & Satyanarayana, S. (2021, March). SMS spam detection using machine learning and deep learning techniques. In *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)* (Vol. 1, pp. 358-362). IEEE.

[10] Alsaffar, D., Alfahhad, A., Alqhtani, B., Alamri, L., Alansari, S., Alqahtani, N., & Alboaneen, D. A. (2019, October). Machine and deep learning algorithms for Twitter spam detection. In *International conference on advanced intelligent systems and informatics* (pp. 483-491). Cham: Springer International Publishing.

[11] Annareddy, S., & Tammina, S. (2019, December). A comparative study of deep learning methods for spam detection. In *2019 third international conference on I-SMAC (IoT in Social, Mobile, Analytics, and Cloud)(I-SMAC)* (pp. 66-72). IEEE.

[12] Rodrigues, A. P., Fernandes, R., Shetty, A., Lakshmanna, K., & Shafi, R. M. (2022). Real-time Twitter spam detection and sentiment analysis using machine learning and deep learning techniques. *Computational Intelligence and Neuroscience, 2022*.

[13] Junnarkar, A., Adhikari, S., Fagania, J., Chimurkar, P., & Karia, D. (2021, February). E-mail spam classification via machine learning and natural language processing. In *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)* (pp. 693-699). IEEE.

[14] Yaseen, Q. (2021). Spam e-mail detection using deep learning techniques. *Procedia Computer Science, 184*, 853-858.

[15] Srinivasan, S., Ravi, V., Alazab, M., Ketha, S., Al-Zoubi, A. M., & Kotti Padannayil, S. (2021). Spam e-mail detection is based on distributed word embedding with deep learning. *Machine intelligence and big data analytics for cybersecurity applications*, 161-189.

[16] Srinivasan, S., Ravi, V., Sowmya, V., Krichen, M., Noureddine, D. B., Anivilla, S., & Soman, K. P. (2020, March). Deep convolutional neural network-based image spam classification. In *2020 6th conference on data science and machine learning applications (CDMA)* (pp. 112-117). IEEE.

[17] Kaddoura, S., Alfandi, O., & Dahmani, N. (2020, September). A spam e-mail detection mechanism for English language text e-mails using a deep learning approach. In 2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE) (pp. 193-198). IEEE.

[18] Abayomi-Alli, O., Misra, S., & Abayomi-Alli, A. (2022). A deep learning method for automatic SMS spam classification: Performance of learning algorithms on indigenous dataset. Concurrency and Computation: Practice and Experience, 34(17), e6989.

[19] Mallampati, D., & Hegde, N. P. (2020). A machine learning based e-mail spam classification framework model: related challenges and issues. International Journal of Innovative Technology and Exploring Engineering, 9(4), 3137-3144.

[20] Sheneamer, A. (2021). Comparison of Deep and Traditional Learning Methods for E-mail Spam Filtering. International Journal of Advanced Computer Science and Applications, 12(1).

[21] Bagui, S., Nandi, D., Bagui, S., & White, R. J. (2021). Machine learning and deep learning for phishing e-mail classification using one-hot encoding. Journal of Computer Science, 17, 610-623.

[22] Karasoy, O., & Ballı, S. (2022). Spam SMS detection for the Turkish language with deep text analysis and deep learning methods. Arabian Journal for Science and Engineering, 47(8), 9361-9377.

[23] Roy, P. K., Singh, J. P., & Banerjee, S. (2020). Deep learning to filter SMS Spam. Future Generation Computer Systems, 102, 524-533.

[24] Sethi, M., Chandra, S., Chaudhary, V., & Dahiya, Y. (2022). Spam E-mail Detection Using Machine Learning and Neural Networks. In Sentimental Analysis and Deep Learning: Proceedings of ICSADL 2021 (pp. 275-290). Springer Singapore.

Appendix

§	
Student ID	
I Supervisor	
Proposed Thesis Title	
Current as at	

#		Project Timeline											
		Month				Month				Month			
		Wk1	Wk2	Wk3	Wk4	Wk5	Wk6	Wk7	Wk8	Wk9	Wk10	Wk11	Wk12
01	**Milestones Key**												
	First Meeting Form Submission												
	Literature Review Draft												
	Submission												
02	**Research proposal and RM lecturer feedback emailed to supervisor**												
03	**Literature review**												
04	**Develop/Source Measures for surveys**												
05	**Develop Recruitment Materials**												
06	**Preparation of surveys/interview schedules**												
07	**Start Recruitment process**												
08	**Methodology: Write Up**												
09	**Pilot Study?**												
10	**Data Collection**												
11	**Analysis**												
12	**Thesis writing/editing**												
13	**Thesis review by supervisors**												
14	**Incorporate feedback**												
15	**Submit Thesis**												
16	**Create X**												
17	**Submit X**												
18													
19													
20													
o	Highlights a key date which you must comply with.												
	Time spent on a task - eg. shade Jan X9 to Jun X9 to indicate you will be working on your research proposal.												
	Expected time frame on task												
NB													

o	This is a sample Key
X	This is a sample Key
Add others you may need	

Samply track or plan using your own key see bottom of table, use fill option you prefer

Sample track or plan using your own key

Fig 33: Gantt Chart