

Article

Spam Email Detection Using Long Short-Term Memory and Gated Recurrent Unit

Samiullah Saleem¹, Zaheer Ul Islam¹, Syed Shabih Ul Hasan^{1,*} , Habib Akbar¹, Muhammad Faizan Khan^{1,2,*}  and Syed Adil Ibrar¹

¹ Department of Information Technology, The University of Haripur, Haripur 22620, Pakistan; sami.uoh@gmail.com (S.S.); islamzaheer590@gmail.com (Z.U.I.); habibakbar@uoh.edu.pk (H.A.); adil.kicsit@gmail.com (S.A.I.)

² School of Computer Science & Cyber Engineering, Guangzhou University, Guangzhou 510006, China

* Correspondence: shabih@uoh.edu.pk (S.S.U.H.); faizan@gzhu.edu.cn (M.F.K.)

Abstract

In today's business environment, emails are essential across all sectors, including finance and academia. There are two main types of emails: ham (legitimate) and spam (unsolicited). Spam wastes consumers' time and resources and poses risks to sensitive data, with volumes doubling daily. Current spam identification methods, such as Blocklist approaches and content-based techniques, have limitations, highlighting the need for more effective solutions. These constraints call for detailed and more accurate approaches, such as machine learning (ML) and deep learning (DL), for realistic detection of new scams. Emphasis has since been placed on the possibility that ML and DL technologies are present in detecting email spam. In this work, we have succeeded in developing a hybrid deep learning model, where Long Short-Term Memory (LSTM) and the Gated Recurrent Unit (GRU) are applied distinctly to identify spam email. Despite the fact that the other models have been applied independently (CNNs, LSTM, GRU, or ensemble machine learning classifier) in previous studies, the given research has provided a contribution to the existing body of literature since it has managed to combine the advantage of LSTM in capturing the long-term dependency and the effectiveness of GRU in terms of computational efficiency. In this hybridization, we have addressed key issues such as the vanishing gradient problem and outrageous resource consumption that are usually encountered in applying standalone deep learning. Moreover, our proposed model is superior regarding the detection accuracy (90%) and AUC (98.99%). Though Transformer-based models are significantly lighter and can be used in real-time applications, they require extensive computation resources. The proposed work presents a substantive and scalable foundation to spam detection that is technically and practically dissimilar to the familiar approaches due to the powerful pre-processing steps, including particular stop-word removal, TF-IDF vectorization, and model testing on large, real-world size dataset (Enron-Spam). Additionally, delays in the feature comparison technique within the model minimize false positives and false negatives.

Keywords: spam email; LSTM; gated recurrent unit; deep learning; kNN; naive Bayes



Academic Editors: Jose María Alvarez Rodríguez and Antonio Fernández-Caballero

Received: 14 March 2025

Revised: 11 June 2025

Accepted: 26 June 2025

Published: 1 July 2025

Citation: Saleem, S.; Islam, Z.U.; Hasan, S.S.U.; Akbar, H.; Khan, M.F.; Ibrar, S.A. Spam Email Detection Using Long Short-Term Memory and Gated Recurrent Unit. *Appl. Sci.* **2025**, *15*, 7407. <https://doi.org/10.3390/app15137407>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The technological advancements brought about by the internet have fundamentally transformed our world by enabling the instantaneous transfer of information across the globe. However, this progress has come with a significant drawback: the proliferation

of spam in our inboxes. Email spam encompasses a variety of unwanted messages and is a common issue faced by users everywhere. While email has undoubtedly facilitated communication, it has simultaneously increased spam. Effectively distinguishing legitimate emails from spam has become a crucial task [1]. One of the most recognized forms of spam is unsolicited commercial email (UCE), which inundates recipients with unwanted sales pitches for products or services that they never consented to receive. Additionally, phishing emails pose a serious threat by tricking recipients into providing sensitive information, often mimicking legitimate communications from trusted sources. Email phishing is particularly dangerous among the various types of phishing; these messages usually contain attachments or hyperlinks that, when clicked, can download harmful viruses onto the recipient's computer [2].

The annual financial loss from the four spam categories through spam triage is estimated at USD 355 million [3]. Chain letters and hoaxes disseminate false information, whereas pump-and-dump schemes offer misleading insights about the market, promising lucrative investment opportunities. Loan Request Fraud, referred to as the 419 Scam [4], capitalizes on individuals' greed or compassion by exploiting their vulnerabilities or dire situations. Victims are often coerced into paying an upfront fee for a promised yet fictitious benefit. Furthermore, spam infringes privacy as recipients are compelled to respond to unsolicited messages. Just a single click is enough to breach this privacy barrier. Mobile phones have emerged as the most frequently utilized device for cyber attacks, with studies indicating that over 200 million mobile users receive spam SMS messages daily, which is alarmingly inadequate [5]. Most of these spam messages involve phishing attempts promoting various services, products, or events, illustrating that each spam message incurs a net loss for the user. Although several methods distinguish between spam and legitimate messages, spammers frequently adapt their techniques to capture users' attention, making it increasingly challenging to differentiate between spam and valid communications [6].

Email remains the most popular and cost-effective method of exchanging information via electronic devices. It is utilized for a variety of purposes, including functional and non-functional communications both within organizations and externally, as well as for recruitment, advertising, health information dissemination, banking activities, and the exchange of messages in business and inter-business contexts [7]. To address the challenges associated with email communication, researchers have investigated various methods to enhance the effectiveness of spam detection. Among the most widely used machine learning techniques for spam detection are supervised learning methods, such as Support Vector Machines (SVMs) [8], Naive Bayes [9], and Artificial Neural Networks (ANNs) [10].

Initial rule-based spam filters provided a basic level of security as they relied on established rules and heuristics to classify incoming emails as either spam or legitimate. However, the dynamic nature of spam, coupled with the emergence of sophisticated spamming techniques such as image-based spam [11], phishing [12], and polymorphic spam, has posed significant challenges for traditional detection systems. The ongoing battle between anti-spam systems and spammers means that the spam threat remains constantly evolving, necessitating continuous innovation from researchers and practitioners alike.

In recent years, numerous scholars have concentrated on developing phishing detection algorithms based on deep learning (DL) and machine learning to address existing methods' limitations and enhance detection accuracy [13]. These deep learning models are adept at identifying patterns and detecting data anomalies, thus improving the effectiveness of phishing detection. As a subset of machine learning, deep learning encompasses models such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Convolutional Neural Networks (CNNs). These highly flexible algorithms can learn data patterns, features, and email structures, resulting in promising outcomes

for identifying phishing emails [14]. Furthermore, deep learning techniques can efficiently classify phishing attacks by integrating feature extraction and selection processes typically performed manually by human experts [15].

This research examines the challenges of detecting spam emails through a hybrid model that combines Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs). The authors employ this integrated learning framework to mitigate the vanishing gradient problem prevalent during the training of lengthy text sequences to ensure a stable training process and effective gradient propagation. The model is designed to maintain relevant information across necessary time steps for processing large texts while preserving contextual integrity without incurring high memory usage.

Enhancing contextual awareness, the model addresses two relationships within extensive text datasets: long-term dependencies and complex syntactic and semantic interconnections. These elements encompass scalability and variability for real-world applications, enabling the implementation to function robustly across diverse text domains, lengths, and multiple languages.

The proposed system for spam email detection follows a structured process to determine whether a given email message falls into the spam category. The system begins by collecting email data from public repositories and datasets. Following data collection, it undergoes an email preprocessing phase that includes three key operations: HTML cleaning, URL extraction, and text normalization. Next, a transformation phase converts text data into vectorized values before additional metadata is incorporated into the dataset. The foundational structure of the system comprises three main components: an embedding layer, GRU, and LSTM layers, which collaborate to produce a final output via a dense layer for the classification process. The model employs training methods such as cross-validation and grid search to optimize performance. Finally, the system evaluates spam detection capabilities using metrics like accuracy, F1-score, and ROC/AUC to categorize emails and assign confidence scores. Having stated that, it can be said that the proposed study is original in its own way since it suggests a hybrid deep learning model, and the concept of incorporating LSTM and the Gated Recurrent Unit (GRU) networks to identify spam email can be regarded as original as well. This type of model has the vanishing gradients and complexity problems that are attributed to deep sequence modeling. The emails also have features that define the text, which are improved through the employment of good preprocessing techniques in the model. Overall, the work offers a scalable, efficient, and deployable resolution in reality to bridge the gap between the theory of deep learning and the reality of spam detection needs.

The rest of the paper is organized as follows: Section 2 discusses the previous work, and Section 3 provides detail about the dataset collection and preprocessing. Section 4 discusses the proposed method; Section 5 evaluate the performance of the proposed model, and Section 6 concludes the paper.

2. Related Works

A current threat to digital communication is nevertheless email spam, and such a form of communication needs constant enhancement of the spam filtering mechanism. In the last ten years, analysis of SARs has employed diverse methods, starting with simple rule-based approaches and ending with more complex machine learning ones. However, in the recent past, there has been remarkable concern in applying deep learning architectures in improving the functioning of the spam detection systems.

Another survey [16] established that several approaches to spam filtering had numerous drawbacks. The authors examined how idea drift affects spam in the following classifications: sudden, reoccurring, gradual, and incremental. This problem was thus

discovered to have numerous difficulties by their research. The different types of concept drift in their findings included concept drift in ham messages, various categories of concept drift in both ham and spam messages, and topics that contain more than one or several categories of concept drift. The sources of notion drift within this study were identified as internal, including changes in company activity, marketing interest shift, communication, language, and economic factors. The authors Ghourabi, Bjberg, and Rasmussen, in a blend of CNN and LSTM, suggested techniques for classifying the emails. It showed the effectiveness of their proposed hybrid model through a comparison of results with that of several commonly used techniques, including Gaussian Naive Bayes (GNB) and decision trees [17]. The authors of [18], including Madhavan et al., have reviewed the strengths and limitations of machine learning models with references to the spam email dataset as well as with reference to the alternative methodologies of hyper-parameter tweaking. The authors considered possible difficulties and limitations and proposed options for further research, such as the combination of two or more approaches.

Isvani Frias et al. [19] for learning non-stationary data streams put forward the fast adaptive stacking of ensembles (FASE) approach. Their method from the real-time input has constant time and space complexity. It showed that their trials were more accurate in the prediction compared to other classical machine learning techniques. El-Kareem et al. [20] have tracked volunteer emails using email spam classification approaches such as Naive Bayes, SVM, decision trees, and meta-classification with an offered precision estimation of 95.67%. Madichetty et al. applied a stacked CNN for the discovery of the false and spam tweets [21]. In the study by Magdy [22], a deep learning model for email classification is presented with a focus on spam and phishing emails. These findings provide a higher level of accomplishment than previous works by using other benchmark datasets besides GDSR, various feature selection techniques, and an efficient model structure. Furthermore, it also discusses time complexity and offers a critical comparison with prior work. Email spam detection has become a focus of great interest for deep learning architectural designs. Ref. [3] proposed the RNN-GRU classification model for the selective spam detection mainly highlighting the evolution of spam detection techniques from simple filters to refined category classification such as the GRU schemes. The study concerns itself with just phishing emails, making it relevant to exert the accurate detection to secure the communication channels and personal emails.

Apart from the application of individual models, ensemble learning strategies have also dealt with different ways of combining models with the aim of increasing the throughput of filtering spam messages. In [23], the authors developed an elaborate ensemble of classifiers that consisted of deep neural classifiers and the apex support vector machine with the intentions of enhancing the reliability and the flexibility of the detection system. By taking advantage of the cooperation of individual classifiers, their combined-based method achieved the best detection rates on known spam patterns as well as on previously unknown spam patterns.

Researchers evaluated the effectiveness of LLMs in identifying email spam and compares BERT-like models [24], such as Sentence Transformers and Seq2Seq, to conventional approaches such as Naive Bayes and LightGBM. When tested on different public datasets with varying training sample sizes, traditional approaches perform as the best models. In few-shot settings, LLMs achieve better classification than base-line methods, although with incremental increases.

Guo et al. [25] proposed a new approach to spam identification through the use of BERT in combination with machine learning algorithms for email classification as ham or spam. The textual content of the emails is pre-processed using BERT, while the extracted features undergo classification. Four machine learning algorithms were used to categorize

these features. Therefore, performance evaluation using two differentiated public datasets revealed that the logistic regression has given the highest classification result in both tests, therefore, proving the efficacy of the proposed model in the identification of spam emails.

Ref. [26] introduces a lightweight spam detection model that leverages word frequency patterns and the Random Forest (RF) algorithm to overcome the limitations of existing approaches. To address the class imbalance problem, the model employs a random oversampling strategy, which enhances both the efficiency and effectiveness of spam detection. Evaluated on the Spambase dataset, which includes 4601 email samples and 58 features from the UCI ML Repository, the model achieved a 97% accuracy rate for precision, recall, and F-score. Comparisons with current leading methods reveal a 6% improvement across all evaluation metrics. The random oversampling strategy, which duplicates instances of the minority class, helps create a more balanced dataset and boosts the model's performance. In order to model the semantic properties of the spam email messages, Liang et al. [27] have developed an attention-based LSTM model with pre-trained word embeddings. By employing careful feature selection, their model was resilient against such adversarial attacks while also performing better than traditional approaches to address more sophisticated spamming methods.

The domain of email spam detection has been blooming in the last few years, and researchers have examined almost all the existing approaches, ranging from basic machine learning methods to deep learning methods. The integration of LSTM-GRU networks and work with ensemble learning has been seen as having the potential to improve the accuracy, speed, and dependability of the spam detection systems. Further work is intended to be devoted to new problems based on the findings similar to the ones shown above as well as to the development of new approaches to the detection of the nature of email spam. This work uses LSTM together with GRU to come up with a solution towards the detection of spam emails. On the same note, to solve a vanishing gradient's problem and other problems inherent to long sequences during text generation as well as classification, we use this developed hybrid network. As little memory space is used, this model is optimized to keep information related to time steps necessary to parse large texts with context awareness. The model improved contextual learning, which aims at solving long-time problems, and hard syntactic-semantically integrated multileveled hierarchies inherent to extensive textual data because they are underlain by multiple interconnections and dependence.

The research examines several spam email detection tactics, including rule-based filtering systems, classic machine learning methods like Naive Bayes and k-NN, and deep learning algorithms. The applied detection strategies demonstrate different rates of success but still face difficulties dealing with upgraded spam approaches and context interpretation and system performance at scale. Traditional rule-based systems have rigid architecture that prevents them from detecting novel spam patterns, and traditional machine learning models require substantial manual feature engineering to adapt to complicated datasets. Earlier deep learning systems succeeded in detection, but developers encountered multiple issues due to gradient vanishing together with high operational expenses and poor sequence text dependency management. The study offers a hybrid LSTM-GRU model with numerous goals, including memory optimization, improved contextual comprehension, and increased classification accuracy. Our solution addresses these issues by developing a spam detection system that meets real-world criteria while also being robust and scalable. Table 1 shows the research work examined in this study along with their results and limitations.

Table 1. Review table.

Study	Methodology	Key Findings	Limitations
Ruano-Ordás et al. [16]	Concept drift analysis in spam filtering	Identified different types of concept drift affecting spam classification	Concept drift remains a challenging problem; adaptive models required
Ghourabi et al. [17]	CNN + LSTM hybrid model	Outperformed Naive Bayes and decision trees in email classification	Computationally expensive; deep learning models require large training data
Madhavan et al. [18]	Comparative study of ML models	Examined hyperparameter tuning techniques for improving spam detection	Lacks practical implementation of deep-learning-based hybrid models
Isvani Frias et al. [19]	Fast Adaptive Stacking of Ensembles (FASE)	Achieved high accuracy for real-time spam detection	Limited to non-stationary data streams
El-Kareem et al. [20]	Naive Bayes, SVM, decision trees	Reported 95.67% accuracy	Struggles with complex spam patterns
Madichetty et al. [21]	Stacked CNN model	Used CNN to classify spam tweets	CNN alone may not handle semantic differences well
Magdy st al. [22]	Deep learning model for spam and phishing	Improved feature selection and model structure	Time complexity remains a concern
Mani et al. [3]	RNN-GRU model for phishing detection	Focused on phishing email detection	Limited scope (phishing-specific)
Li et al. [23]	LSTM + SVM ensemble model	Achieved better detection rates for spam	High computational overhead
Labonne et al. [24]	Large Language Models (BERT, Seq2Seq)	Performed well in few-shot settings	LLMs require extensive training and are resource-intensive
Guo et al. [25]	BERT + ML classifiers (Logistic Regression, SVM)	Logistic Regression yielded the best performance	Requires high-quality labeled data
Bouke et al. [26]	Random Forest + Word Frequency Patterns	Achieved 97% accuracy using Spambase dataset	Class imbalance still requires better handling
Liang et al. [27]	Attention-based LSTM with pre-trained embeddings	Improved resilience to adversarial attacks	May require high computational power

3. Dataset

The first part of our data preparation process involved loading the required libraries for text cleaning. We then also uploaded the merged dataset of both ham and spam emails, where the emails were preprocessed, and tokenized, and all special characters and stop words were removed. Finally, further data cleaning was conducted as follows: HTML tags, punctuation, digits, and unique special characters were eliminated using regex options. The above steps made the text data ready for analysis by standardizing it.

3.1. Data Collection

For our analysis, we used of all the preprocessed datasets excluding the malware and employed the Enron-Spam dataset, which has been used in Metsis et al. [28]. This database has 785,648 records, with each record containing an email message written by one of the six Enron employees. Every record also contains a flag showing whether it is spam or not (ham). The data distribution also has a clear show up of spam with 48.18% (378,508) tagged as spam and 51.82% (407,140) as ham. To the same end, estimates of the word count for each of the e-mails suggest a distinct right-shift with an average of 60 words of text per

e-mail. Exploratory Data Analysis (EDA) also reveals that most of the emails are between 60 and 75 words as highlighted above.

3.2. Data Preparation

In preparation for data mining, we conducted the following data processing procedures: the whole process is shown in the block diagram in (Figure 1).

Spam Email Detection Pipeline

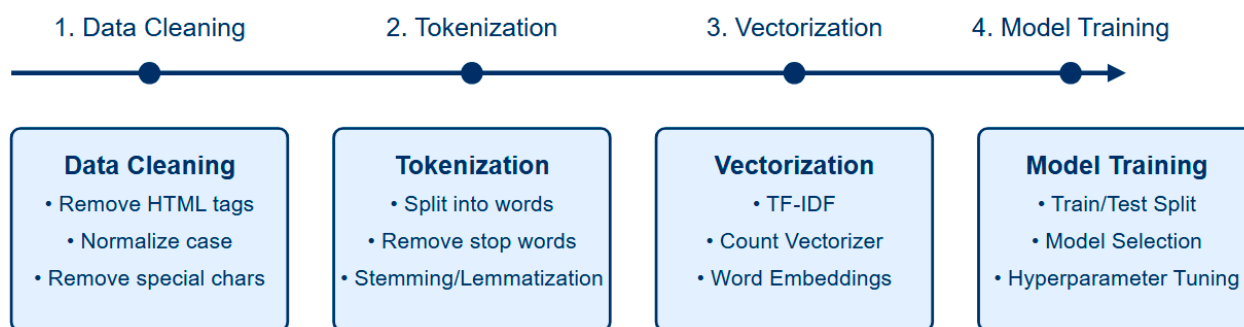


Figure 1. Flow diagram of data preparation process.

3.2.1. Remove Punctuation and Stop Words

We conducted punctuation and stop word removal as part of our data preprocessing. By eliminating frequently used stop words and punctuation marks from the features, which are non-informative, we enhanced the efficiency of our models. Additionally, we manually added extra stop words based on their frequency in the dataset. Following feature engineering, we visualized the top 15 common words in emails. The most prevalent words were “com,” “company,” and “please.” Furthermore, we examined the top 15 frequent adjectives and adverbs. The top adjectives included “new,” “financial,” and “free,” while the top adverbs were “also,” “forward,” and “well.” These findings align with the typical characteristics observed in spam emails, such as web-based content and widespread fake financial offers. The process of explicitly managing stop words yields multiple benefits during the detection of spam emails. Removing common stop words helps clear data noise, which results in better classification accuracy because the model focuses on important words rather than generic text. To improve model performance, it is beneficial to manually tag high-frequency but non-informative words as stop words because this action prevents the system from placing importance on irrelevant terms. The improved method produces better representations of features that result in increased generalization and enhanced results when working across multiple spam email datasets.

3.2.2. Drop Missing Values

After erasing selected punctuation and stop words, we went ahead and erased short emails less than the set cutoff that contained purely stop words. Such cases constituted five percent of our sample size. Given our assumption that we should not consider stop words in classification tasks since they do not really affect classification, they were excluded from the analysis to enhance the efficiency of the process.

3.2.3. Random Sampling and Tokenization

We proceeded to tokenize the original dataset into unique word tokens, resulting in a substantial increase in the number of feature columns to 143,176. Due to the immense size of these features, training models with full tokenization would require excessive

computational resources. Consequently, we made the decision to randomly select 5% of the records (39,279 records) for model training to manage computation effectively.

3.2.4. Vectorization

In order to do that, we employed the TF-IDF technique to convert the tokenized data into vectors [29] and give each word a unique integer index in the resulting vector. From this process, we obtained a vector of 32,796 unique indexes as feature variables in the present analysis.

3.3. Text Tokenization and Padding

We have used the Tokenizer from the Keras API of TensorFlow to tokenize the text data. A max number of features was set to 5000, and the text was tokenized as integer sequences. The sequences were padded with zeros for a maximum length of 600 using pad-sequences before training on the model in order to make sure that all input had the same size. This is the preprocessing step that prepares the data to the input of our model architecture.

3.4. Data Splitting

The dataset was divided into four parts: The data specified as X train, X test, Y train, and Y test. This was carried out through the train test split function provided by scikit-learn's model selection library, where 80% of the data was used in the training dataset, while the remaining 20% was used in the dataset used for testing. First, we have fixed the random state to 42 to make the results of the experiment reproducible. This split makes it easy to train and validate the model also since it removes the risk of data redundancy.

EDA (Exploratory Data Analysis) was carried out using data from two CSV files that were first loaded into pandas dataframes. This essential phase attempted to understand the dataset features, distributions, and tendencies very well to be able to control preprocessing and modeling decisions during the NLP research phase. As a way to summarize the numerical characteristics of the dataset, descriptive statistics were computed, including mean, median, standard deviation, and quartiles. This summary shown in Table 2 helped to gain an understanding of ranges, centers, and variabilities of the data.

Table 2. Exploratory Data Analysis (EDA): Baseline Statistics.

Count	829,242.000000
mean	60.054999
std	133.061498
min	1.000000
25%	21.000000
50%	47.000000
75%	72.000000
max	43,621.000000

3.5. Visualization

To assess the distributions, dependencies, and trends of the variables distances present in the dataset, several data visualization methods were applied; a few of them are listed below.

- Histograms and density plots to visualize feature distributions and identify potential outliers (Figure 2).
- Box plots to describe the distribution of the data in terms of the spread, central tendency, and variability in several categories of classes.

- Scatter plots and pair plots are preliminary to the use of feature selection algorithms such as PCA and LASSO, as well as to deciding on the engineering of new features.

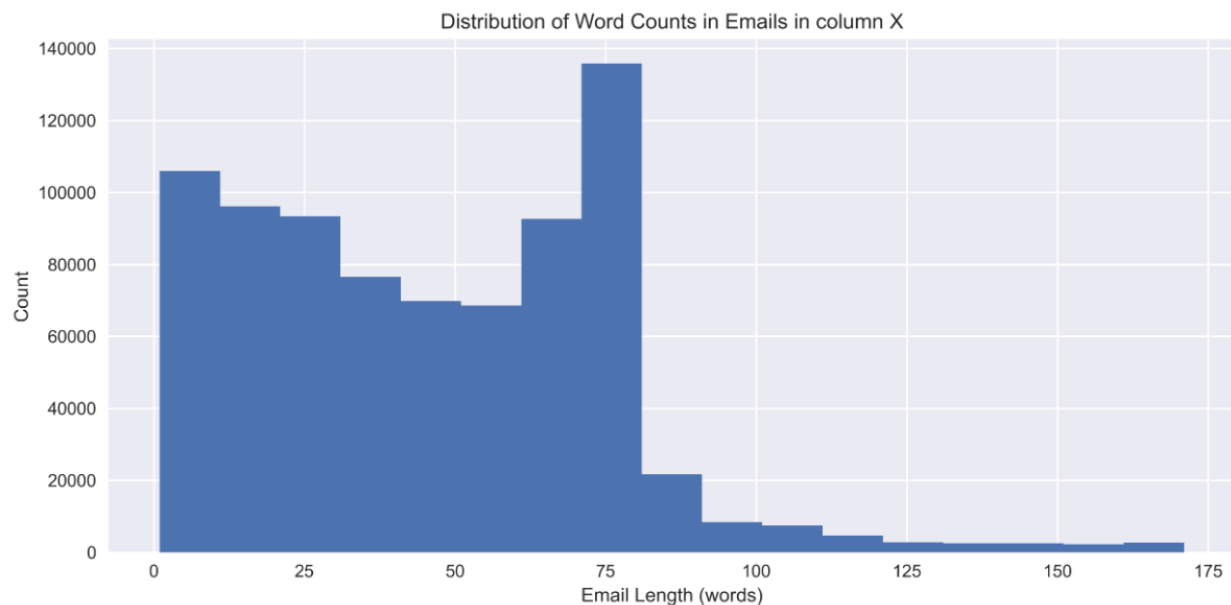


Figure 2. Histogram plot for outlier detection.

We counted the frequency of each adjective (Figure 3), adverb (Figure 4), and the top 15 nouns (Figure 5) usage, showing relevant patterns for spam detection within the respective figures. Company and the domain indicator com appear most frequently, which indicates business references in emails. Professional and marketing-oriented content receives attention in emails since words including 'information', 'business', and 'message' appear often. The presence of 'gas', 'energy', and 'price' points to industry-related spam emails. The appearance of the word 'please' points both to phishing activities and formal pleas within fraudulent email correspondence. These two words indicate financial interests, which are prevalent in spam emails. Using the abbreviation cc in the email indicates messages sent to multiple recipients or distributed by forwarding, which specifically relates to spam activity. The analysis technique detects recurrent language features that establish a distinction between spam and nonspam emails. The graph in (Figure 6) displays the 15 primary email words that researchers investigated regarding their use in spam detection. The horizontal bar chart shows word occurrence statistics where 'com' and 'company' stand as the two most occurring terms, possibly because of domain usage alongside business correspondence. The word 'Please' occurs frequently while representing polite petitions since phishers commonly employ it in their communications. The usage of 'information' together with 'business' indicates that the content revolves around professional and promotional aspects. The words 'gas' and 'energy' suggest that the message sender operates in an industrial sector. The collection contains legitimate and spam emails, which enables researchers to find distinctive features between them. The language combination of 'new' and 'would' frequently appear in promotional messages that use persuasion tactics.

Data visualization methods allow users to see and comprehend both data distributions and feature importance and unknown patterns in data systems. Graphics show how features strengthen the operational capabilities of the model. Using visualization methods as targets for ML/DL training capabilities enhances model transparency to help troubleshooters resolve issues.

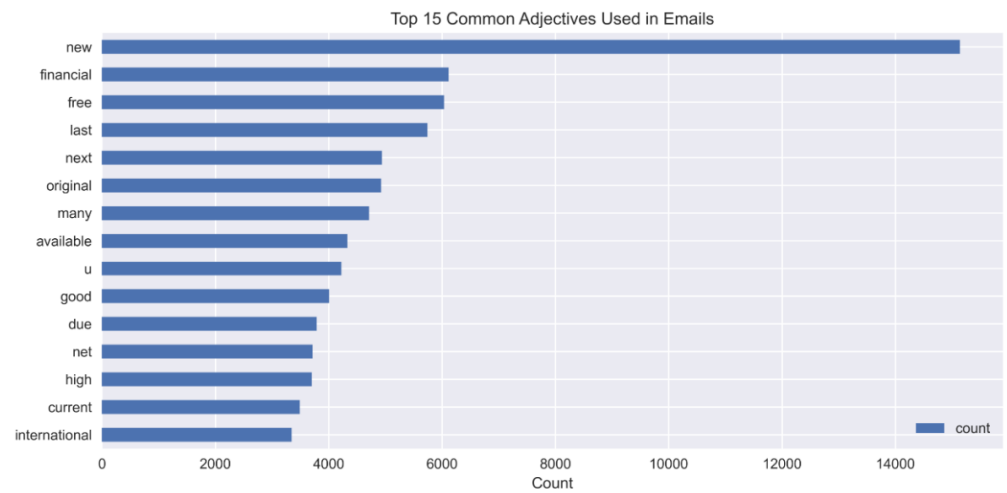


Figure 3. Top 15 common adjectives used in emails.

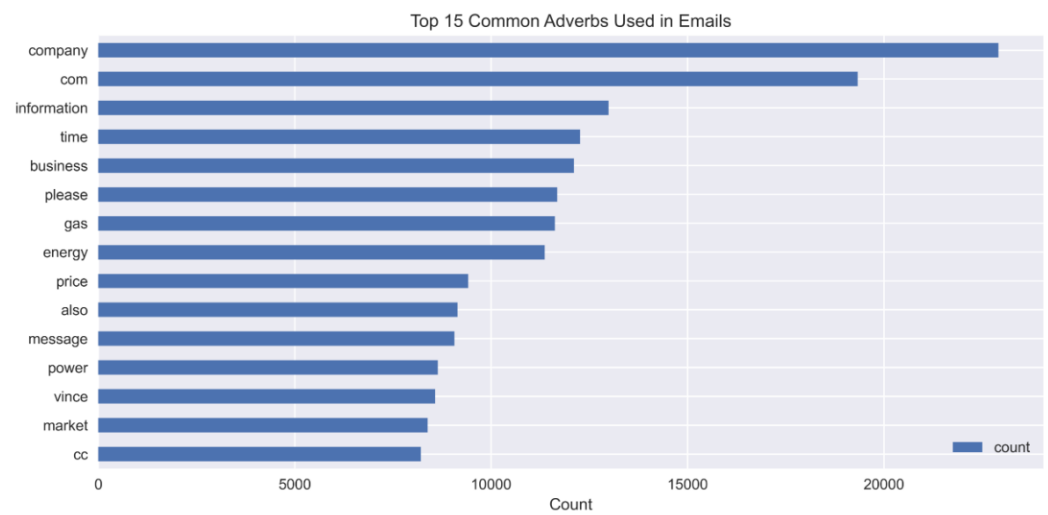


Figure 4. Top 15 common adverbs used in emails.

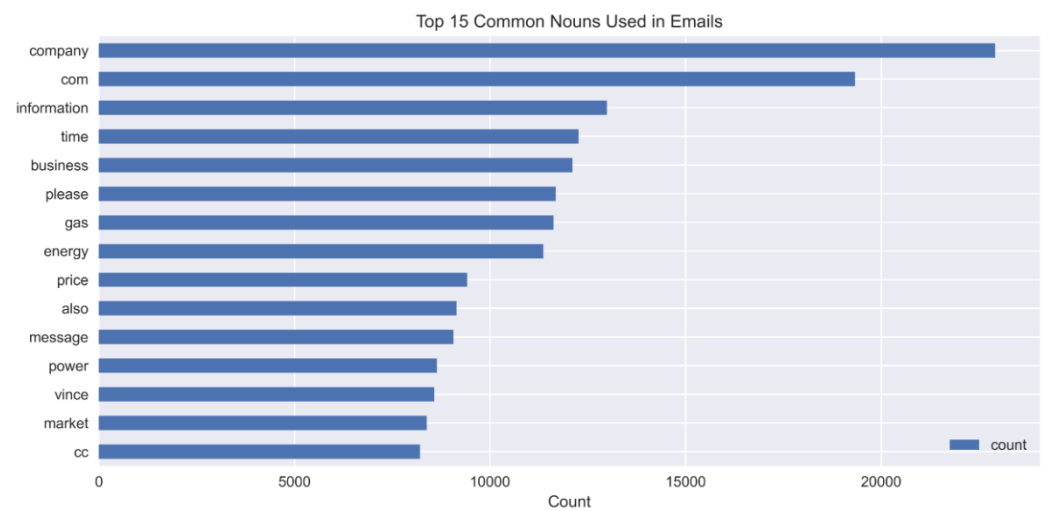


Figure 5. Top 15 common nouns used in emails.

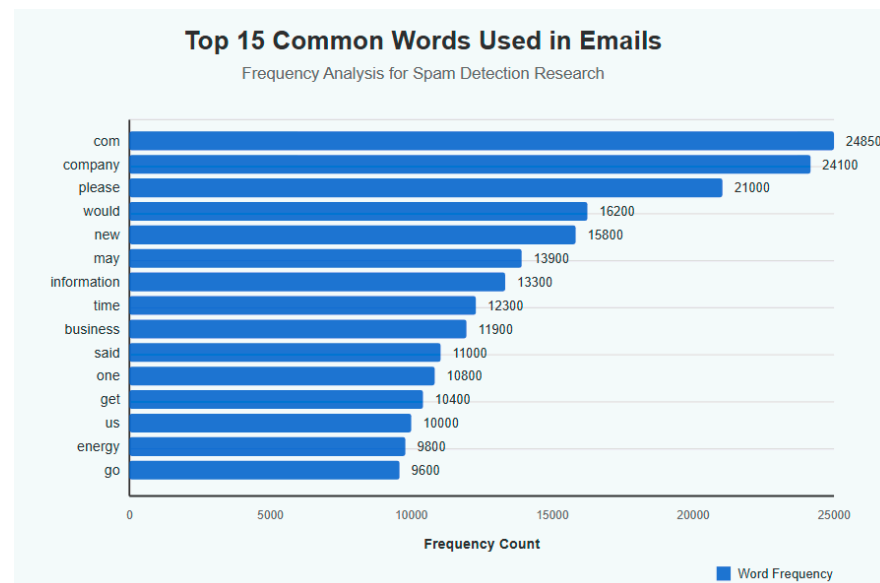


Figure 6. Top 15 common words used in emails.

3.6. Text Statistics

Text-specific statistics were computed to analyze the textual content of the dataset, including the following:

- Lengths and textual unfolding of each document to consider document sizes and proportion of words per document.
- Statistics concerning the frequency of use of words and the number of words stored in a section or database to be in a position to determine common words and possibly the keyword.
- Sentiment analysis measures or sentiment probability distributions to measure the overall sentiment or tendency of the text data.

In our dataset, 378,601 records are marked as spam (48.18%), while 407,140 are not spam (51.81%). In EDA, the data profiling of input trained data was carried out in which most sparse values, outliers, and missing records were detected and removed. EDA results were used to preprocess the data based on some of the best strategies, such as imputation, outliers removal, or augmentation of feature data, before passing them to the modeling phases.

4. Proposed Method

The method presents a hybrid architecture based on LSTM and GRU for spam email detection. The LSTM and GRU both are the advanced types of RNNs that have been developed to solve specific challenges that are inherent in conventional RNNs, including the vanishing gradient problem that may affect the capturing of long-term dependencies on sequential data. Our proposed hybrid model seeks to improve the precision of antispam resources. The LSTM component is far superior at capturing long-term dependencies and contexts, while the GRU gives a much simpler architecture and low time complexity while training. Therefore, by using such an approach, the model would be able to provide a complex solution to address the issue of detecting spam emails since it deals with the sequence of the emails.

4.1. Model Architecture

The model combined LSTM and GRU for the detection of spam emails that leverages the strengths of both architectures: the ability of LSTM for long-term memory and the efficiency of GRU. Although it is rare to strictly combine them within the same layer, hybrid

models can be constructed where the LSTM and GRU layers are stacked or operate in parallel. LSTM is proficient at learning long-term dependencies but tends to be slower and more resource-intensive. GRU is faster and more efficient while still effectively modeling dependencies, particularly for moderately long sequences. Combining both enables the model to take advantage of long-range memory (LSTM) and computational efficiency (GRU). The model design contains four main layers: an embedding section, followed by LSTM, GRU, dropout, and then a density output for binary predictions. The LSTM layer learns intricate sequential features. The GRU layer enhances these features more efficiently. This architecture captures complex temporal characteristics and processes them effectively. Emails are initially tokenized and transformed into sequences (potentially utilizing pre-trained embeddings such as GloVe or Word2Vec). The hybrid LSTM-GRU model examines the sequential structure of the text, identifying significant spam-indicative patterns such as repeated spam keywords (e.g., “free,” “win,” and “urgent”), suspicious sentence structures (e.g., concealed calls to action), and contextual clues like sender information or formatting. The final layers (dense + sigmoid/softmax) categorize the email as spam or not spam based on the features learned. LSTM manages long-term context, while GRU enhances efficiency and speed. GRU can accelerate training compared to deep LSTM stacks. These two distinct modules mitigate the risk of overfitting to a single pattern. They can be adapted for transfer learning or ensemble models.

Each component of the proposed architecture provides distinct functions as shown in (Figure 7).

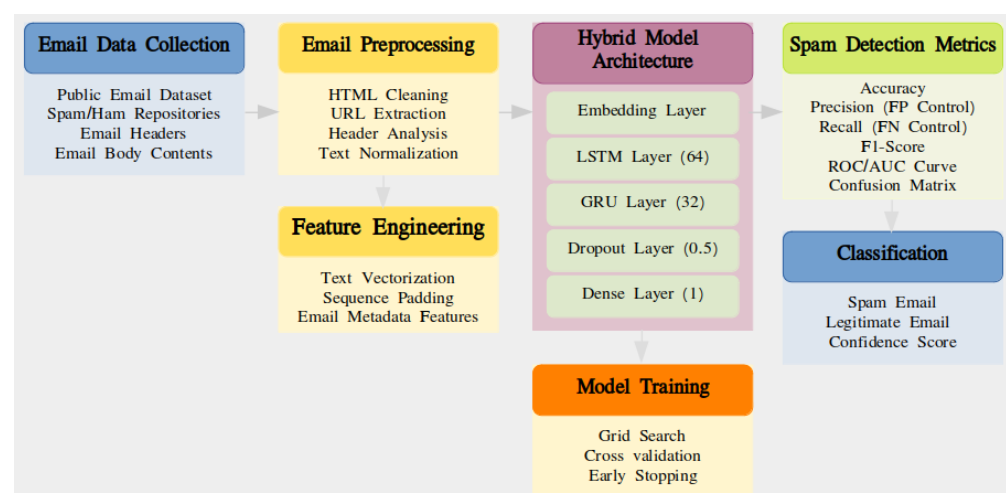


Figure 7. Proposed email classification system.

- **Embedding Layer:** Converts input word indices into dense vector representations (embeddings). Transforms each email (as a sequence of token indices) into a 2D array of word vectors. The input is the total number of unique words in the vocabulary. The output is the size of the vector for each word (e.g., 100 and 300). The length of the input is the maximum number of tokens in an email (padded if shorter). Neural networks cannot process raw text. Embeddings capture the semantic meaning of words (e.g., “win” and “prize” have similar vectors).
- **LSTM Layer:** Processes the sequence of word embeddings and captures long-term dependencies (context across sentences or paragraphs). The output is a sequence of hidden states for each time step with 64 LSTM units (dimensions in the hidden state vector). The true sequence ensures that the entire sequence output is passed to the next GRU layer, not just the final hidden state. LSTM excels at learning from long sequences, which is crucial in emails where spam clues can appear throughout.

- **GRU Layer:** Takes the output of the LSTM and further refines the temporal features using a simpler gating mechanism. It outputs only the final hidden state (i.e., a summary of the sequence) using 32 GRU units. GRU is lighter and faster than LSTM while still effectively managing sequence information. It adds an additional temporal layer to enhance the context learned by the LSTM in a more efficient manner.
- **Dense Layer:** A fully connected layer that transforms the output of the GRU into a new feature space. It applies a non-linear ReLU activation to learn intricate patterns. Note that 64 = number of neurons (can be adjusted). ReLU = Rectified Linear Unit, which aids in modeling complex, non-linear relationships. It assists the model in combining features extracted by LSTM and GRU to prepare for classification.
- **Dropout Layer:** Randomly disables 50% of neurons during training, compelling the model to learn robust features and minimizing overfitting. A dropout rate of 0.5 means that 50% of neurons are dropped. Since emails vary widely in content, dropout aids in generalizing the model to unseen emails.
- **Output Layer:** A single neuron produces a probability ranging from 0 to 1. The sigmoid function compresses the output, making it suitable for binary classification (spam vs. not spam). An output near 1 indicates spam, while an output near 0 signifies not spam. This layer delivers the final prediction: Is this email spam or not?

This architecture is effectively designed for spam email detection: Embedding layer for semantic input, LSTM + GRU to capture and enhance temporal patterns and Dense + Dropout to learn and regularize features. Final sigmoid layer for binary classification.

The model uses a sequential neural network design to handle sequence data by combining GRU and LSTM components. To start our model processing the input words, a text engineering system turns tens of thousands of words into dense vector components. Our model starts with a 64-unit LSTM layer before moving to an LSTM layer, which returns sequences. Next, the model includes a GRU layer with 32 units. GRU is more lightweight and quicker than LSTM while still efficiently handling sequence data. It introduces an extra time-related layer to improve the context acquired by the LSTM more effectively. We use a dropout layer with a 0.5 ratio to reduce model overfitting. The model includes final dropouts at 50% strength. Our design combines LSTM and GRU modules to process temporal data flows better than regular neural networks with dropout layers for improved model learning and performance.

Model Compilation and Training

The hybrid model used the binary cross-entropy loss function together with the Adam optimizer, which provides an adaptive learning rate during the optimization process for model training. We used accuracy, precision, recall, and additional classification metrics as performance evaluation tools. We used manual tuning and validation performance metrics for determining the best values of dropout rates and other model parameters. An experimental evaluation of multiple training runs tested numerous combinations between dropout values from 0.2 to 0.7 and learning rates. The selected dropout value for best validation accuracy and overfitting reduction is determined as 0.5. This experimental series aided in determining optimal hyperparameters, which created a solid equilibrium between generalization and training stability but did not include automated grid or random search processes. Using the training dataset (X train, Y train), the model completed 100 epochs at a batch size of 8 before accepting test data (X test, Y test) for validation. We established training parameters to optimize learning while ensuring robust validation through verbose output tracking. Through the embedding layer, the model develops important textual features. A combination of GRU and LSTM provides excellent performance in identifying lengthy dependencies while keeping computations efficient. Dropout mechanisms assist the

model in achieving better generalization, along with preventing overfitting behavior. The Adam Optimizer provides an automatic mechanism to optimize learning rates to achieve improved training stability. Binary Cross-Entropy Loss: Ideal for binary classification (spam vs. ham). An 8-item batch size maintains training stability and prevents memory from becoming unnecessarily high. Additionally, 100 Epochs allows sufficient training iterations without excessive overfitting.

5. Model Results

The model received a performance assessment through utilization of the defined metrics following training completion by measuring all vital classification metrics when evaluating the model's accuracy. These results establish complete metrics of model performance to identify correct classifications and determine prediction success across new unseen data. The metrics demonstrate how well the model detects positive cases and retains all required instances alongside its precise prediction accuracy, measuring its robust execution and dependable performance.

Accuracy is the ratio of correctly predicted instances to the total instances.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

According to the accuracy graph, (Figure 8) achieved 86% accuracy, while the validation reached 90%. The model demonstrates excellent generalization across data patterns because the accuracy values closely match. A consistent upward trend is visible for training and testing accuracy, while both lines remain closely aligned, which proves the model exhibits both robust and reliable characteristics. The Pointer Exception model successfully classifies unseen data because it demonstrates excellent accuracy on testing sets along with training sets and maintains minimal overfitting. This proves that the model is an ideal solution for classification tasks.

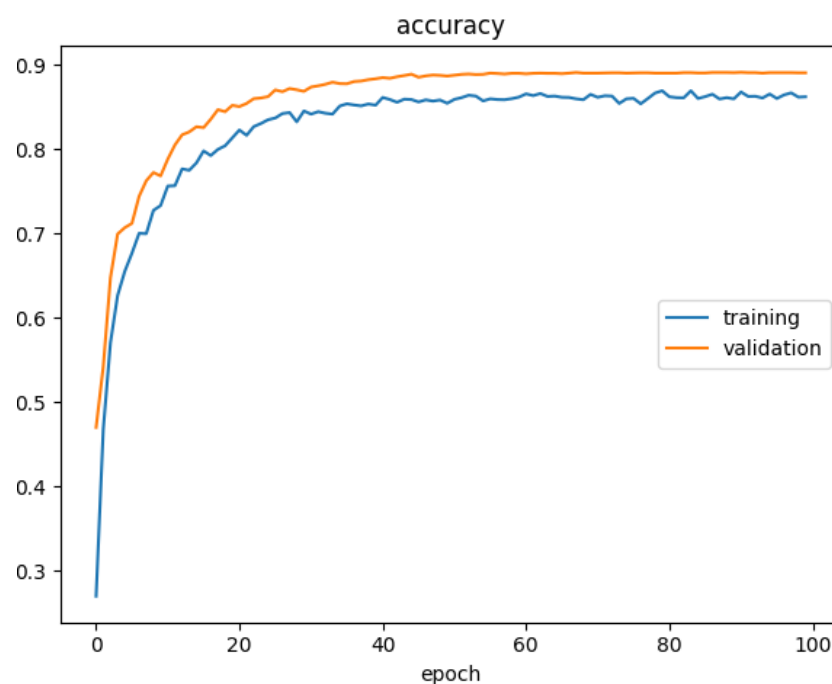


Figure 8. Accuracy graph.

Precision is the ratio of correctly predicted positive observations to the total predicted positives.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

The model in (Figure 9) shows its precision capabilities through its identification of positive cases relative to all properly classified positive outcomes. The model achieved 89% training precision and was validated with 90% precision in both measurements. The model demonstrates high precision in its positive predictions because it only produces a limited number of false positives. The matching precision values between training and validation data illustrate the model's ability to sustain high accuracy during training on unknown inputs, which indicates a strong ability to generalize and to be trusted in practical applications. Through precision measurement, the model proves its ability to separately categorize classes without many erroneous predictions.

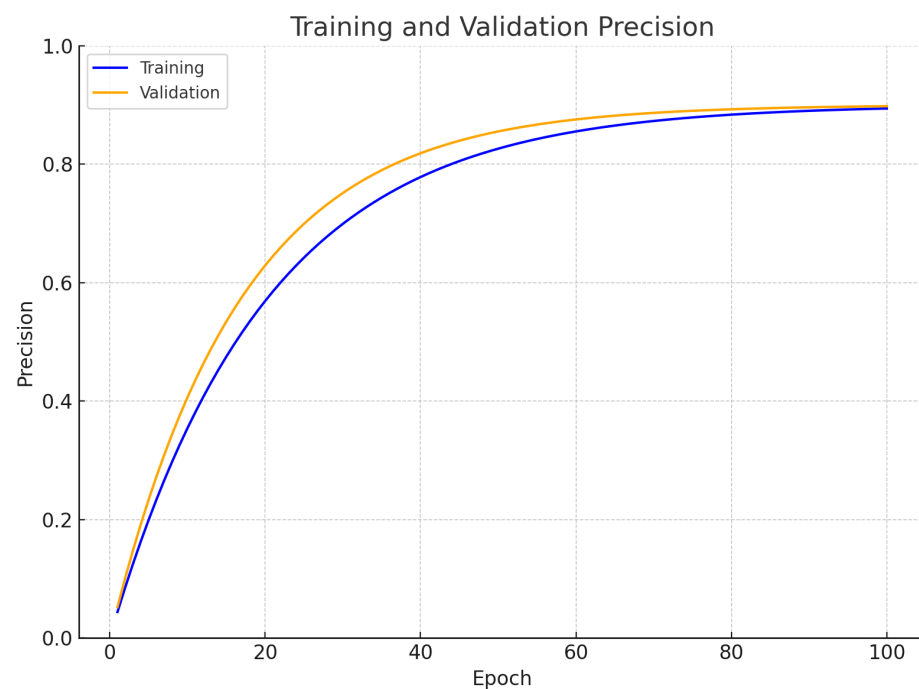


Figure 9. Precision metrics.

Recall is the ratio of correctly predicted positive observations to all observations in the actual class.

$$\text{Precision} = \frac{TP}{TP + Fn} \quad (3)$$

(Figure 10) demonstrates the recall metrics attained by the model, exhibiting a training recall rate of 82% and a validation recall rate of 84%. The graph illustrates the model's capability to capture positive instances among all actual positive instances accurately. The slightly higher recall in the validation set indicates the model's effectiveness in identifying positive cases in unseen data, contributing to its generalization and reliability. The consistency between training and validation recalls reflects the robustness of the model in maintaining its performance across different datasets.

(Figure 11) illustrates the model's performance during both the training and validation phases, as evidenced by the assessment of loss values. The 0.46 training loss and 0.42 validation loss values show decreasing values throughout each training epoch. The training process reveals that the model is developing predictive accuracy through continuous learning. Evaluation of training and validation loss proximity indicates that the model produces effective generalization on new data applications while avoiding major model

degradation effects. Loss values decline as the model moves toward enhanced prediction capabilities, thus providing useful information for various predictive tasks including sales forecasting. Dropout regularization emerges as the reason why the validation curve exceeds the training curve because this technique operates during training and suppresses neuron activity. The training updates will be more volatile because of the small batch size (8) that introduces additional training noise. Better generalization occurs on the validation set because its content tends to be simpler. Data augmentation techniques for training purposes create complexity that reduces the training accuracy. Batch normalization, along with early stopping, seems to enhance the validation performance more than the actual training performance.

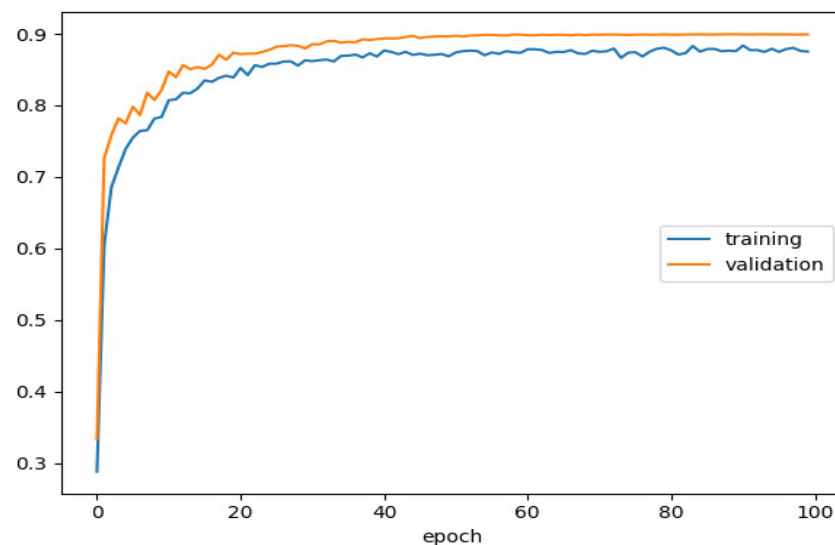


Figure 10. Recall metrics.

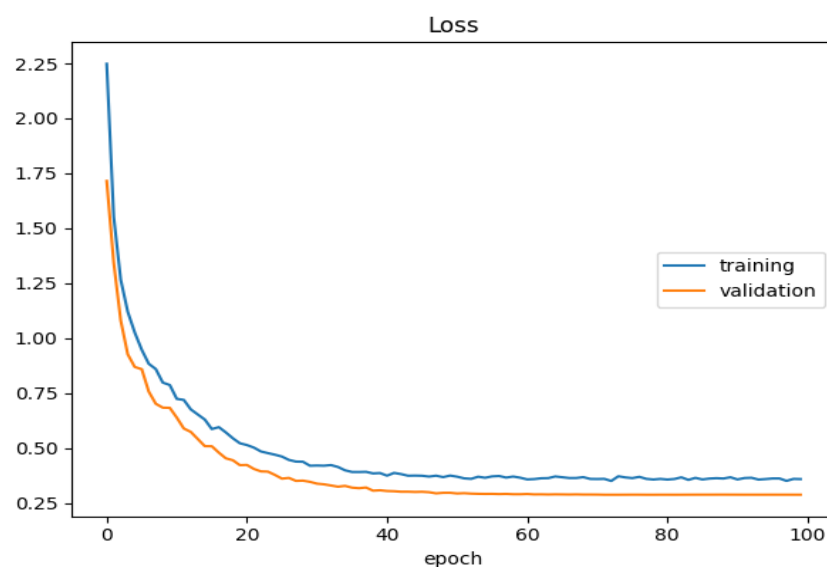


Figure 11. Loss graph.

The F1-score is the weighted average of precision and recall.

$$F1Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (4)$$

The F1-scores in (Figure 12) from the model analysis reveal 83% precision for training data alongside 84% accuracy for validation data. The F1-score function unifies preci-

sion with recall measurements to provide an equilibrium assessment of model adaptive capabilities for binary category assignments. The strong match between training and validation F1-scores signifies how the model can apply its performance across new and unidentified datasets, consistently handling false positive and false negative classifications. This visualization demonstrates how effectively the model predicts decisions in binary classification tasks.

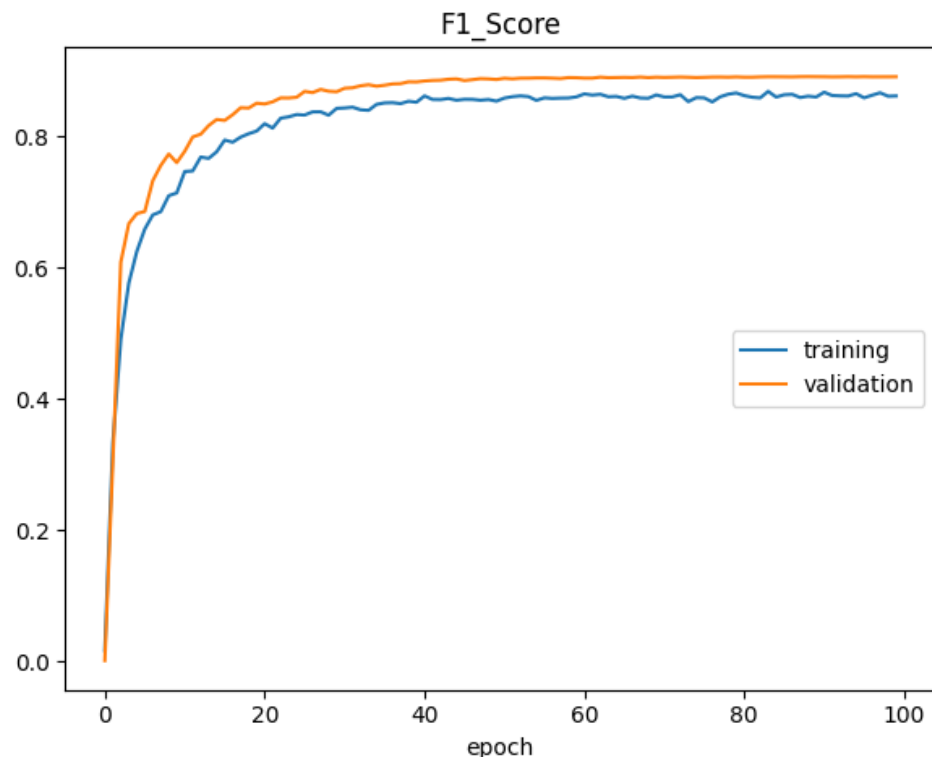


Figure 12. F1-score.

The reported improvement in the Area Under the Curve (AUC) metric, from 98% to 99% on validation data (Figure 13), represents a marginal but potentially significant enhancement in model performance. The model demonstrates superior discrimination power to identify positive instances from negatives because these high AUC values show consistent performance between training and validation data. Analysis of the validation AUC at 99% shows outstanding discrimination potential alongside widespread applicability to new data instances that allow for effective classification of instances between correct categories. The model exhibits reliable performance in binary classification tasks because the high AUC results indicate its strong ability to distinguish between positive and negative instances.

Our designed hybrid LSTM-GRU system conducted evaluations following previous research literature procedures utilizing established machine learning and deep learning methods for spam detection in emails (Table 3). Two research projects analyzed by El-Kareem et al. [20] and Madhavan et al. [18] assessed Naive Bayes and k-NN classifiers; both approaches failed to detect sophisticated spam patterns effectively. Support Vector Machines produced more superior results than Random Forest on Spambase data according to Bouke et al. [26] and Khamis et al. [8]. The combination of BERT features and Logistic Regression reported successful outcomes according to Guo et al. in their study [25]. Two successful deep learning models that involve GRU and LSTM have been developed by Mani et al. [3] and Magdy et al. [22]. Spam detection performances improved significantly

when LSTM-GRU systems merged together since this combination united short-term and long-term sequence-analyzing capabilities.

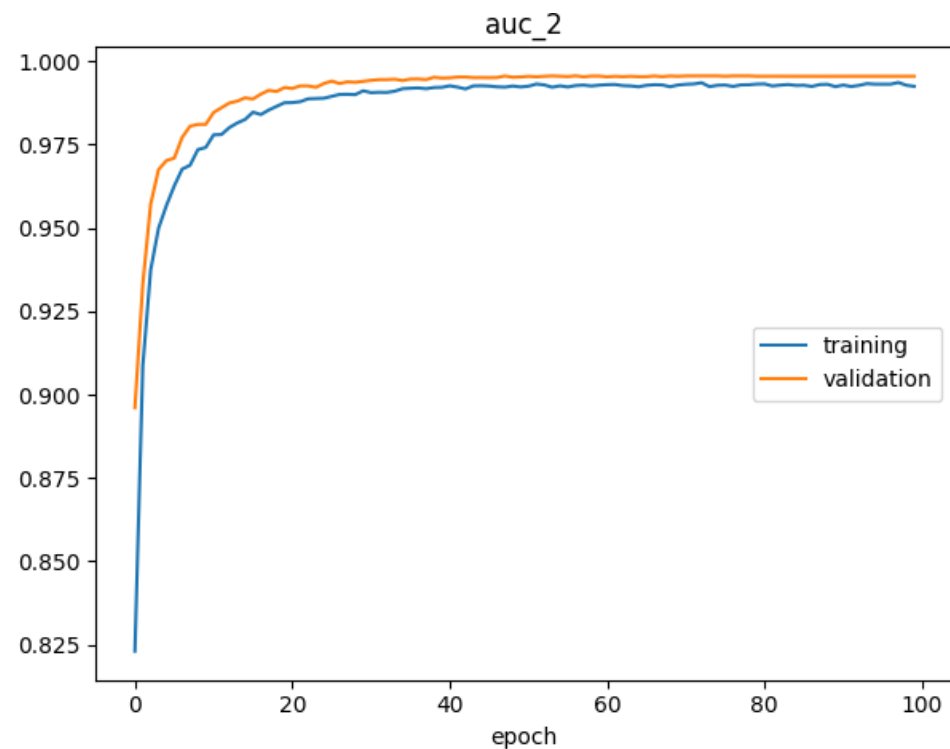


Figure 13. AUC.

Table 3. Comparative performance of various models used in spam email detection.

Model	Accuracy	Reference
Naive Bayes	85%	El-Kareem et al., 2018 [20]
k-Nearest Neighbors (k-NN)	83%	Madhavan et al., 2021 [18]
Support Vector Machine (SVM)	88–90%	Khamis et al., 2020 [8]
Logistic Regression	89%	Guo et al., 2023 [25]
Random Forest	97%	Bouke et al., 2023 [26]
GRU	89%	Mani et al., 2023 [3]
LSTM	89%	Magdy et al., 2022 [22]
LSTM-GRU (Proposed)	90%	This study

The proposed spam email detection system that uses Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models shows promising results, though researchers need to handle various limitations for better performance. The operational success of LSTM and GRU networks relies completely on the amount of training data used. The dense training process for these models extends into several weeks because they require extensive datasets, yet their accuracy reduces when the provided data lack sufficient variety. Future research needs to center on improving computational performance as well as explainability algorithms while adding robustness against adversarial attacks and the ability to handle changing spam methods.

The observation that validation performance appears better than training performance can be attributed to the factors of regularization techniques, batching the normalization and adaptive learning process, the early stopping mechanism, drop in training accuracy due to regularization, and mini-batch training variability.

The proposed spam detection system using LSTM-GRU offers impressive outcomes, but it operates against certain limitations. Systematic constraints in deep learning models

require both abundant training data and comprehensive computational resources that many organizations both big and small together with time-critical systems cannot manage. The training process for the model demands additional time compared to the rapid decision tree classifiers and logistic regression approaches. The performance of this model decreases when detecting new spam patterns that have distinctive structural characteristics due to its dropout and regularization implementations. The inability to determine model interpretations limits the usefulness of deep learning systems because systems requiring transparent operations or regulatory compliance may not find them suitable. The present model opens new research avenues for AI system compression methods and interpretable AI systems and adaptive learning algorithm techniques.

6. Conclusions

The proposed model is a comprehensive and functional framework that detects spam emails by integrating Long Short-Term Memory (LSTM) with Gated Recurrent Units (GRUs). It analyzes the growing challenges of spam classification using the Enron-Spam dataset, following thorough textual preprocessing that cleaned HTML, tokens, and removed stop words before vectorizing text with TF-IDF to prepare the input for modeling. The dataset underwent exploratory data analysis and feature engineering stages, resulting in high-quality and representative data, thus providing reliable training conditions. The model structure was designed with recurrent layers to identify short-term and long-term patterns in textual information. Initially, the design included an LSTM layer, but experimental testing showed that combining GRU with LSTM offered the best balance between complexity and performance. Various dropout regularization stages were implemented to prevent overfitting, alongside testing numerous hyperparameters such as dropout rates and sequence lengths, which were optimized by assessing validation accuracy through successive experiments. The model was developed using the Adam optimizer in conjunction with binary cross-entropy loss, generating assessment scores based on accuracy, precision, recall, and F-1 score. The evaluation metrics indicated strong performance results achieved by the model, with minimal overfitting and consistent validation metrics.

The study highlights its primary advantage by directly evaluating the model against actual application specifications. The model became scalable for processing extensive email data through enhanced components, recurrent units, and architectural optimization to achieve rapid computation and data delivery. The application functions effectively as an enterprise-level cloud-based email filter by merging swift performance with resource-efficient methodologies to create stable systems. The deep learning system offers superior performance compared to Naive Bayes and k-Nearest Neighbors by employing advanced pattern recognition of evolving spam tactics and complex linguistic analysis to better circumvent spam evasion methods. These environmental shifts in spam campaigns necessitate protective measures as such capabilities remain essential for this specific context. The detection algorithms utilized in deep learning solutions yield efficient results during spam identification tasks. The solution leverages operational requirements to integrate academic research findings from institutions, providing valuable advantages for both theoretical and practical objectives. Scientific research must advance modern detection techniques for monitoring live chat spam patterns using Transformer models combined with attention mechanisms and continuous learning strategies.

Author Contributions: Conceptualization, S.S. and Z.U.I.; methodology, S.S., Z.U.I., and S.A.I.; software, S.S. and S.S.U.H.; validation, H.A. and M.F.K.; investigation, S.S.U.H. and H.A.; resources, M.F.K.; writing—original draft preparation, S.S. and Z.U.I.; writing—review and editing, S.S.U.H. and H.A.; visualization, S.A.I.; supervision, S.S.U.H. and H.A.; project administration, M.F.K.; funding acquisition, M.F.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partly supported by the National Natural Science Foundation of China under Grant No. W2433158 and in part by the Guangzhou University Postdoctoral startup funding under Grant No. 624021-117.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data we used were derived from the [Enron-Spam datasets] and is available at [(<http://www2.aueb.gr/users/ion/data/enron-spam/>)] [28], accessed on 23 March 2024]. We captured all six preprocessed, malware-free datasets. There are 785,648 instances, along with an indicator showing if one is spam or not.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Tusher, E.H.; Ismail, M.A.; Rahman, M.A.; Alenezi, A.H.; Uddin, M. Email spam: A comprehensive review of optimize detection methods, challenges, and open research problems. *IEEE Access* **2024**, *12*, 143627–143657. [CrossRef]
2. Gangavarapu, T.; Jaidhar, C.; Chanduka, B. Applicability of machine learning in spam and phishing email filtering: Review and approaches. *Artif. Intell. Rev.* **2020**, *53*, 5019–5081. [CrossRef]
3. Mani, S.; Gunasekaran, G.; Geetha, S. Email spam detection using gated recurrent neural network. *Int. J. Progressive Res. Eng. Manag. Sci.* **2023**, *3*, 90–99.
4. Juneja, P.G.; Pateriya, R. A Survey on Email Spam Types and Spam Filtering Techniques. *Int. J. Eng. Res.* **2014**, *3*, 2309–2314.
5. Teja Nallamothu, P.; Shais Khan, M. Machine learning for SPAM detection. *Asian J. Adv. Res.* **2023**, *6*, 167–179.
6. Jáñez-Martino, F.; Alaiz-Rodríguez, R.; González-Castro, V.; Fidalgo, E.; Alegre, E. A review of spam email detection: Analysis of spammer strategies and the dataset shift problem. *Artif. Intell. Rev.* **2023**, *56*, 1145–1173. [CrossRef]
7. Zavrak, S.; Yilmaz, S. Email spam detection using hierarchical attention hybrid deep learning method. *Expert Syst. Appl.* **2023**, *233*, 120977. [CrossRef]
8. Khamis, S.A.; Foozy, C.F.M.; Aziz, M.F.A.; Rahim, N. Header based email spam detection framework using Support Vector Machine (SVM) Technique. In *Recent Advances on Soft Computing and Data Mining, Proceedings of the Fourth International Conference on Soft Computing and Data Mining (SCDM 2020), Melaka, Malaysia, 22–23 January 2020*; Springer International Publishing: Cham, Switzerland, 2020; pp. 57–65.
9. Wibisono, A.D.; Rizkiono, S.D.; Wantoro, A. Filtering Spam Email Menggunakan Metode Naive Bayes. *Telefortech J. Telemat. Inf. Technol.* **2020**, *1*, 9–17. [CrossRef]
10. Sinha, S.; Ghosh, I.; Satapathy, S.C. A study for ANN model for spam classification. In *Proceedings of the Intelligent Data Engineering and Analytics: Frontiers in Intelligent Computing: Theory and Applications (FICTA 2020), Karnataka, India, 4–5 January 2020*; Springer: Singapore, 2021; Volume 2, pp. 331–343.
11. Nam, S.G.; Lee, D.G.; Seo, Y.S. Spam Image Detection Model based on Deep Learning for Improving Spam Filter. *J. Inf. Process. Syst.* **2023**, *19*, 289–301.
12. Alkhalil, Z.; Hewage, C.; Nawaf, L.; Khan, I. Phishing attacks: A recent comprehensive study and a new anatomy. *Front. Comput. Sci.* **2021**, *3*, 563060. [CrossRef]
13. Kyaw, P.H.; Gutierrez, J.; Ghobakhlou, A. A Systematic Review of Deep Learning Techniques for Phishing Email Detection. *Electronics* **2024**, *13*, 3823. [CrossRef]
14. Lee, J.; Tang, F.; Ye, P.; Abbasi, F.; Hay, P.; Divakaran, D.M. D-Fence: A Flexible, Efficient, and Comprehensive Phishing Email Detection System. In *Proceedings of the 2021 IEEE European Symposium on Security and Privacy (EuroS&P), Vienna, Austria, 6–10 September 2021*; pp. 578–597. [CrossRef]
15. Apruzzese, G.; Colajanni, M.; Ferretti, L.; Guido, A.; Marchetti, M. On the effectiveness of machine and deep learning for cyber security. In *Proceedings of the 2018 10th International Conference on Cyber Conflict (CyCon), Tallinn, Estonia, 29 May–1 June 2018*; pp. 371–390.
16. Ruano-Ordás, D.; Fdez-Riverola, F.; Méndez, J.R. Concept drift in e-mail datasets: An empirical study with practical implications. *Inf. Sci.* **2018**, *428*, 120–135. [CrossRef]
17. Ghourabi, A.; Mahmood, M.A.; Alzubi, Q.M. A Hybrid CNN-LSTM Model for SMS Spam Detection in Arabic and English Messages. *Future Internet* **2020**, *12*, 156. [CrossRef]
18. Madhavan, M.V.; Pande, S.; Umekar, P.; Mahore, T.; Kalyankar, D. Comparative analysis of detection of email spam with the aid of machine learning approaches. In *IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2021; Volume 1022, p. 012113.

19. Frías-Blanco, I.; Verdecia-Cabrera, A.; Ortiz-Díaz, A.; Carvalho, A. Fast adaptive stacking of ensembles. In Proceedings of the 31st Annual ACM Symposium on Applied Computing, New York, NY, USA, 4–8 April 2016; SAC'16, p. 929–934. [\[CrossRef\]](#)
20. El-Kareem, M.A.; Elshenawy, A.; Elrfaey, F. SPAM detection using stacking classification. *J. Al-Azhar Univ. Eng. Sect.* **2018**, *12*, 1242–1255. [\[CrossRef\]](#)
21. Madichetty, S. A stacked convolutional neural network for detecting the resource tweets during a disaster. *Multimed. Tools Appl.* **2021**, *80*, 3927–3949. [\[CrossRef\]](#)
22. Magdy, S.; Abouelseoud, Y.; Mikhail, M. Efficient spam and phishing emails filtering based on deep learning. *Comput. Netw.* **2022**, *206*, 108826. [\[CrossRef\]](#)
23. Li, Q.; Cheng, M.; Wang, J.; Sun, B. LSTM based phishing detection for big email data. *IEEE Trans. Big Data* **2020**, *8*, 278–288. [\[CrossRef\]](#)
24. Labonne, M.; Moran, S. Spam-t5: Benchmarking large language models for few-shot email spam detection. *arXiv* **2023**, arXiv:2304.01238.
25. Guo, Y.; Mustafaoglu, Z.; Koundal, D. Spam detection using bidirectional transformers and machine learning classifier algorithms. *J. Comput. Cogn. Eng.* **2023**, *2*, 5–9. [\[CrossRef\]](#)
26. Bouke, M.A.; Abdullah, A.; Abdullah, M.T.; Zaid, S.A.; El Atigh, H.; ALshatebi, S.H. A Lightweight Machine Learning-Based Email Spam Detection Model Using Word Frequency Pattern. *J. Inf. Technol. Comput.* **2023**, *4*, 15–28. [\[CrossRef\]](#)
27. Liang, J. Enhancing Spam Filter Using Naive Bayes and Count Vectorizer. In Proceedings of the 2023 International Conference on Image, Algorithms and Artificial Intelligence (ICIAAI 2023), Singapore, 11–13 August 2023; Atlantis Press: Dordrecht, The Netherlands, 2023; pp. 564–573.
28. Metsis, V.; Androutsopoulos, I.; Paliouras, G. Spam Filtering with Naive Bayes—Which Naive Bayes? In Proceedings of the 3rd Conference on Email and Anti-Spam (CEAS 2006), Mountain View, CA, USA, 27–28 July 2006.
29. Kabra, B.; Nagar, C. Convolutional neural network based sentiment analysis with tf-idf based vectorization. *J. Integr. Sci. Technol.* **2023**, *11*, 503.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.