IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 August 2024, accepted 19 September 2024, date of publication 25 September 2024, date of current version 11 October 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3467996

TOPICAL REVIEW

Email Spam: A Comprehensive Review of Optimize Detection Methods, Challenges, and Open Research Problems

EKRAMUL HAQUE TUSHER[1], MOHD ARFIAN ISMAIL[1,2], MD ARAFATUR RAHMAN[3], (Senior Member, IEEE), ALI H. ALENEZI[4], AND MUEEN UDDIN[5], (Senior Member, IEEE)

[1]Faculty of Computing, Universiti Malaysia Pahang Al-Sultan Abdullah, Pekan, Pahang 26600, Malaysia
[2]Center of Excellence for Artificial Intelligence and Data Science, Universiti Malaysia Pahang Al-Sultan Abdullah, Gambang 26300, Malaysia
[3]School of Mathematics and Computer Science, University of Wolverhampton, WV1 1LY Wolverhampton, U.K.
[4]Remote Sensing Unit, Electrical Engineering Department, Northern Border University, Arar 73213, Saudi Arabia
[5]College of Computing and Information Technology, University of Doha for Science and Technology, Doha, Qatar

Corresponding authors: Mohd Arfian Ismail (arfian@umpsa.edu.my) and Mueen Uddin (mueen.uddin@udst.edu.qa)

This work was supported in part by the Fundamental Research Grant (FRGS) with FRGS/1/2022/ICT02/UMP/02/2 from the Ministry of Higher Education Malaysia under Grant RDU220134; in part by Qatar National Library—QNL (Open Access Research); and in part by the Deanship of Scientific Research at Northern Border University, Arar, Suadi Arabia, under Project NBU-FFR-2024-2159-07.

ABSTRACT Nowadays, emails are used across almost every field, spanning from business to education. Broadly, emails can be categorized as either ham or spam. Email spam, also known as junk emails or unwanted emails, can harm users by wasting time and computing resources, along with stealing valuable information. The volume of spam emails is rising rapidly day by day. Detecting and filtering spam presents significant and complex challenges for email systems. Traditional identification techniques like blocklists, real-time blackhole listing, and content-based methods have limitations. These limitations have led to the advancement of more sophisticated machine learning (ML) and deep learning (DL) methods for enhanced spam detection accuracy. In recent years, considerable attention has focused on the potential of ML and DL methods to improve email spam detection. A comprehensive literature review is therefore imperative for developing an updated, evidence-based understanding of contemporary research on employing these methods against this persistent problem. The review aims to systematically identify various ML and DL methods applied for spam detection, evaluate their effectiveness, and highlight promising future research directions considering gaps. By combining and analyzing findings across studies, it will obtain the strengths and weaknesses of existing methods. This review seeks to advance knowledge on reliable and efficient integration of state-of-the-art ML and DL into identifying email spam.

INDEX TERMS Email spam, machine learning, deep learning, fuzzy system, feature selection, spam detection.

I. INTRODUCTION

Emails have become an essential component of the contemporary lifestyle, which is heavily influenced by technology. Since its introduction to the public in the mid-1990s, the use of emails has had a noticeable positive effect on various sectors such as business, healthcare, education, and industry. Emails have facilitated collaboration among individuals by offering a cost-effective and expeditious mode of communication [1]. They have greatly facilitated communication and information exchange on both personal and professional levels. However, the increasing usage and reliance on emails have also exposed users to greater cybersecurity risks in the form of spam attacks, malware infections, and other modes of exploitation [2]. As emails continue to play a pivotal role across domains, it is critical for users as well

The associate editor coordinating the review of this manuscript and approving it for publication was Parul Garg.

as organizations to adopt safe email practices and robust security measures against emerging threats. Cybercriminals utilize email channels as a launchpad for assaults that have the potential to seriously hurt both people and organizations. Indeed, it is claimed that emails are responsible for as much as 90% of cyberattacks [3]. Even though there have been efforts to improve email security, vulnerabilities still exist. For the purpose of exploiting organizations and compromising their systems, attackers utilize a variety of strategies, such as social engineering, hacking email accounts, and the fabrication of bogus emails [4]. Social engineering initiatives are among the most misleading of these tactics since they are designed to trick personnel, accomplish unauthorized access, disclose sensitive information, disseminate malware, and disrupt essential activities [5]. It is therefore absolutely necessary to take action against these growing dangers that are based on email and to boost cybersecurity prevention measures [6]. There are vulnerabilities in email networks that are routinely exploited by malicious individuals. The most common methods of attack that these individuals use are spam and phishing emails.

Emails have mostly made conversation and connection easier, but a big problem is that people keep getting spam emails. Segregating legitimate emails from unwanted spam has therefore become a critical task. Studies show that spam accounts for over 50% of global email traffic [7], with healthcare and dating scams being highly common. The volume of spam is rising in line with the overall growth in emails worldwide. By 2025, an estimated 376 billion emails will be sent daily, to over 4.6 billion users [8]. This torrent of spam incurs significant economic and social costs. From consuming network resources to jeopardizing privacy, dealing with spam leads to major technical and infrastructure expenditures [9]. Additionally, research indicates the frustration from spam can negatively impact mental well-being [10].

Every day, more than 320 billion unsolicited emails are produced, and this method is utilised to disseminate 94% of malicious software. The projected financial impact was estimated to be $12 billion due to the dissemination of unsolicited commercial emails to corporate email recipients [11]. Figure 1 from Statista's report on January 16, 2023 shows that a large amount of spam emails were sent globally on that particular day. Approximately 8.6 billion emails were received by the United States, with the Czech Republic and the Netherlands following with 7.7 and 7.6 billion emails, respectively. [8].

There are five primary categories of spam: mobile spam, messaging spam, e-mail spam, search engine optimization (SEO) spam, and social networking spam. Figure 2 provides an overview of prevalent categories of e-mail spam. Based on a virus analysis, it was determined that 94% of malware was transmitted by email. A majority of spam emails have an attachment, with approximately 45% of these attachments being Office document files. Windows programmers ranked second, accounting for 26% of virus transmission through spam e-mail [8].

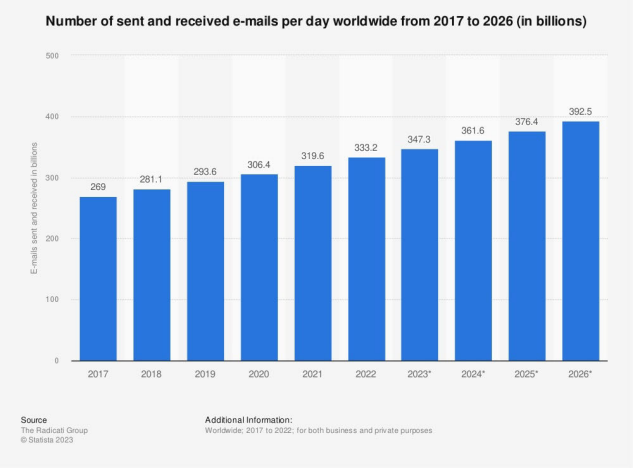

FIGURE 1. Worldwide everyday spam emails [8].

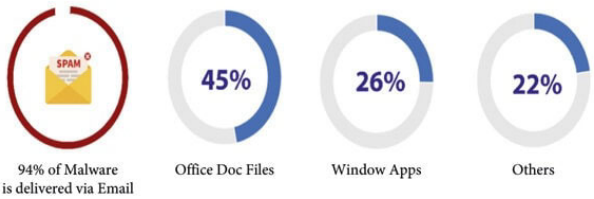

FIGURE 2. Common types of spam email [8].

According to the secure list report, Figure 3 reveals that Russia is the foremost country in terms of outgoing spam, accounting for 23.5% of the total. Germany follows closely in second place with 11%, while the United States ranks third with 10.85% [12].

Efficient and secure digital communication relies heavily on the detection of email spam. Efficient spam detection preserves users from undesired and potentially dangerous emails that can cause time wastage, resource consumption, and jeopardize personal or corporate data. Email systems boost user experience, increase productivity, and protect against security concerns like phishing and malware by effectively filtering out spam. Researchers have suggested many approaches and techniques, including the utilisation of Real-Time Blackhole List [13], Blocklist [14], and Content-Based Filters [15], to detect and eliminate spam from legitimate messages for more than twenty years. Ongoing research is currently being performed to design techniques that are more efficient and precise. Specifically, researchers have shown significant interest in artificial intelligence (AI)-based approaches in recent years [16]. Significant attention has been given to the ML based methods [7], [17], [18]. Moreover, spam email detection has lately witnessed successful implementation of DL methods [19], [20], [21]. The results of these studies demonstrate that ML and DL methods offer an efficient framework for effectively addressing the issue of spam identification, but they also suffer difficulties such as managing incorrect positive and negative results, adjusting

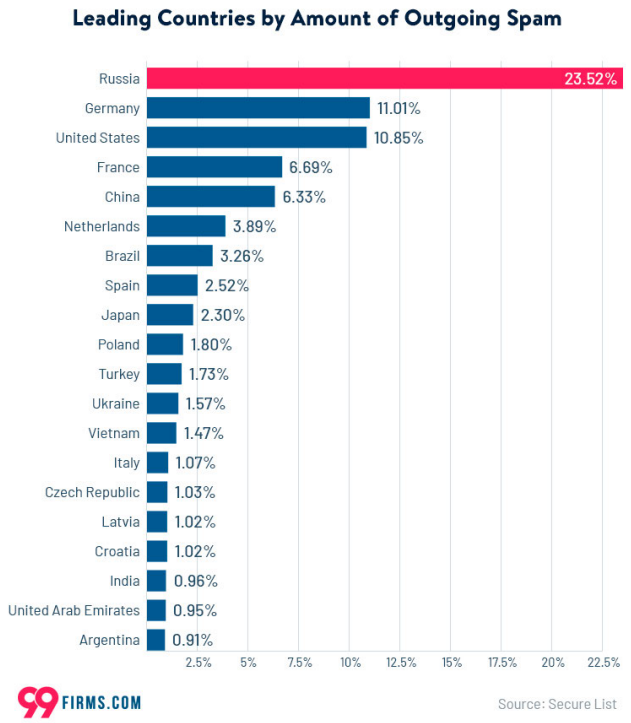

FIGURE 3. Leading countries in sending spam [12].

to novel spam strategies, and guaranteeing computational effectiveness. Furthermore, maintaining a proper balance between precision and the utilization of resources continues to be a crucial concern. Researchers and practitioners in the field face an ongoing challenge to keep ahead of spammers as they constantly improve their techniques [22]. However, there is still potential for additional enhancement and advancement.

The use of ML and DL to find spam is an area that is changing quickly and has gained a lot of attention lately because it has the potential to get around the problems with traditional methods and make detection more accurate. But there needs to be a comprehensive overview of the present research in this field. This can help us figure out the pros and cons of different ML and DL methods and guide the growth of future research. It is possible to get a clear and evidence-based view of how ML and DL can be used to find spam through gathering together the results of all the relevant studies and finding any gaps in the literature. In particular, this kind of study can show how well different ML and DL methods work, as well as their flaws and possible ways to make them better. A thorough study can also find gaps in the research and help with coming up with new research questions and areas to focus on.

A. REVIEW SCOPE

In our comprehensive investigation of more than one hundred research publications sourced from renowned scientific databases such as IEEE Xplore, Web of Science, ScienceDirect, and Scopus, we discovered a noteworthy pattern. There were a number of surveys that addressed the more general topic of email spam; however, there was a significant void in the literature that explicitly focused on detection approaches. Recognising that there has been very little attention paid to the identification of spam in email, our objective was to make a significant contribution by limiting the scope of our investigation and presenting a comprehensive analysis of the most effective ML and DL techniques that are currently being used in this area. Our work places an emphasis on the most recent developments in email spam detection methods, with a particular focus on specialised and optimised approaches. Table 1 offers a comparative analysis between existing survey papers on ML and DL applications in the broader email spam domain and our focused exploration of detection techniques. Through this paper, we sought to illuminate every potential application of ML and DL methods in email spam detection, presenting a comprehensive overview of the field. Furthermore, we have meticulously highlighted the impacts of optimized methods and addressed the challenges of scaling up these innovative solutions within spam detection systems. By doing so, we aim to provide researchers and practitioners with a thorough understanding of the current landscape, potential improvements, and future directions in email spam detection using advanced ML and DL techniques.

B. CONTRIBUTION

There are gaps in understanding the effectiveness, limitations, and potential improvements needed for current spam detection techniques. With the field evolving rapidly and accurate identification of spam being crucial, an updated comprehensive review is needed. This review would synthesize available evidence on existing methods, highlight literature gaps to address through new research, and provide the following key contributions:

- The paper presents a comprehensive review of the crucial characteristics used to identify email spam, as well as significant advancements in this field. The survey identifies significant research gaps and outlines future research goals in the field of email spam detection, based on a comprehensive analysis of existing literature.
- This review paper focuses on the various ML and DL methods utilized for spam email detection and analyzes the effectiveness of existing techniques in accurately identifying spam messages.
- The review presents an elaborate study of several methods applied to email spam detection over the period 2005-2024.
- Analyses the performance of ML and DL methods by examining the findings reported in recent research. Presents a concise summary of these findings in well-organised tables.
- The review identifies the strengths and limitations of various spam detection methods. Analysing the current

TABLE 1. Summary of previous reviews in email spam detection.

Previous Reviews	Email Spam	ML	DL	Feature Selection	Architecture	Dataset Corpus	Period Covered
[13]	✓	✓	×	×	×	✓	1994 - 2013
[14]	✓	✓	×	×	✓	✓	2004 - 2013
[15]	✓	✓	×	×	×	✓	2002 - 2014
[7]	✓	✓	×	×	✓	✓	2000 - 2018
Our Methods	✓	✓	✓	✓	✓	✓	2005 - 2024

literature highlights the key challenges that need to be solved to improve the accuracy and efficacy of identifying spam emails.
- Focuses the scope on an understudied niche area of ML and DL methods in email spam detection to fill a literature gap and make novel contributions.

Overall, this review paper offers valuable insights by presenting a concentrated technical summary, performance evaluation, and future prospects primarily aimed at email spam detection. The discoveries are intended to accelerate progress in this promising domain.

The structure of this paper is as follows: Section II provides an overview of the existing literature and discusses the results obtained from the survey. Section III offers a comprehensive examination of the prominent methods employed for the identification of email spam. Section IV provides an overview of the data pre-processing and specific information regarding the datasets that are accessible to the public. Section V for the implication of the research. Section VI presents the challenges that were observed during the research. Section VII presents the research gaps and open research problems. The conclusion is presented in Section VIII. Figure 4 below shows the paper's structure, which should help readers grasp it better and make the paper easier to read:

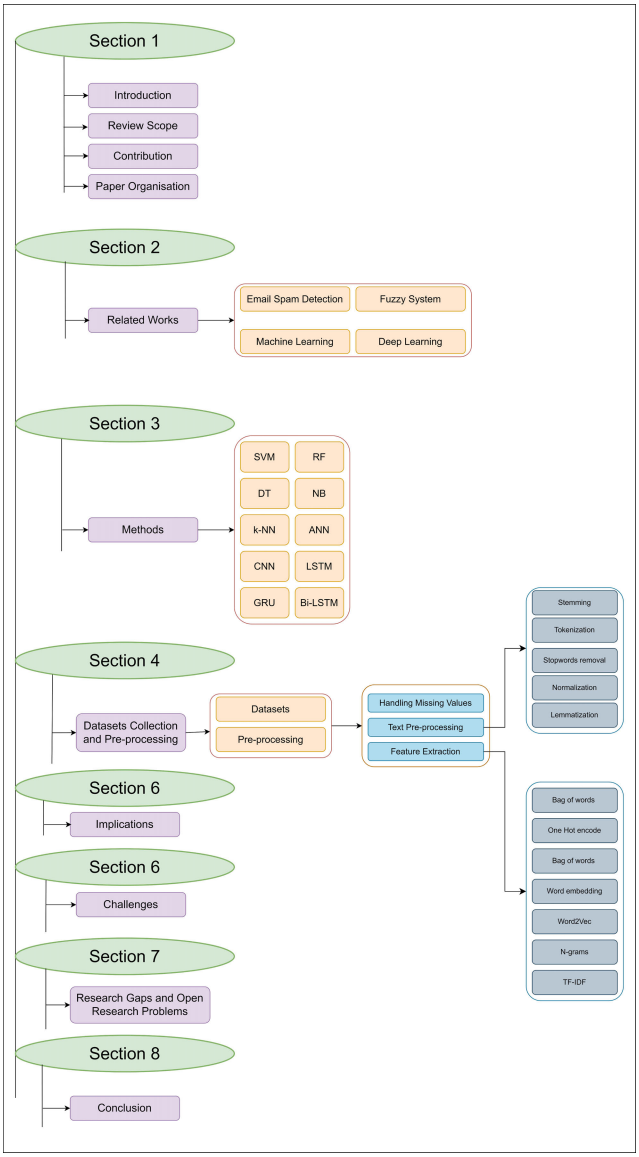

FIGURE 4. A Visualization of the organization of this paper's structure.

II. RELATED WORKS

A. EMAIL SPAM DETECTION

Email refers to electronic mail sent from one device to another over the internet. Since innovating online communication in the early days of networking, email continues playing a vital role in both personal and professional realms despite competition from messaging apps and social media [23]. Since its inception, email has evolved to become a versatile and indispensable tool in both personal and professional spheres [24]. Figure 5 presents three major email service providers that most people utilize - Gmail, Yahoo, and Outlook. Each email platform has distinct advantages and is better suited for certain use cases over the others.

Gmail, offered by Google, is one of the most widely adopted email services globally. Benefits of using Gmail include high inbox storage capacity, excellent search functionality, seamless integration with other Google Workspace apps like Drive and Calendar, and robust spam detection powered by artificial intelligence. The ads-supported model enables providing these features free of cost [25]. Gmail is ideal for personal communication and works well with most email clients.

Yahoo Mail is also a popular free email platform. Key advantages include custom domains to maintain a professional brand, disposable email addresses to protect identity, automatic data download in case of account hacking, and seamless communication tools like chat and SMS

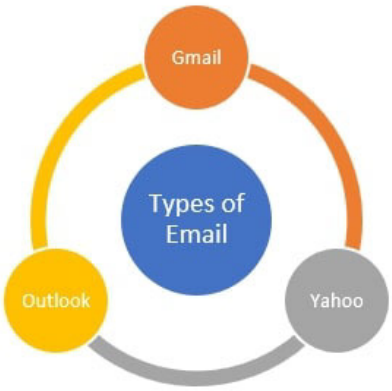

FIGURE 5. Types of the email.

built-in with the interface [26]. These features make Yahoo Mail suitable for business use cases like newsletters, mass communication and privacy protection.

Outlook refers to the email client offered by Microsoft, often bundled with Office suite subscriptions. It works great with Microsoft apps like Word, Excel and Teams. Outlook provides top-notch calendar organization features, robust communication and collaboration functionalities for enterprises, and high security including encryption that complies with financial and healthcare regulations [27]. These capabilities make Outlook popular among corporations and businesses.

The capacity to serve edu mail is a connection across all of these email providers. These emails facilitate administrative duties, course management, and group research by ensuring secure communication within academic communities. Edu mail helps users develop a professional identity by providing access to educational materials, software discounts, and increased privacy precautions. In contrast to Yahoo's simplicity and ease of use, Gmail interacts with Google Workspace for Education. Outlook, which is part of Microsoft 365, offers powerful technologies like OneDrive and Teams [28]. Based on the specific requirements of the institution, each platform improves edu mail with features like enhanced collaboration tools, security measures, and intuitive interfaces.

The ability to exchange thoughts and ideas has increased as communication has developed over time. From the time when communication was limited to face-to-face interactions, letter writing, phone conversations, and text messaging to the present, online presence, communication has changed and become more affordable. Email is a helpful communication tool with a wider audience. There are two categories of emails: ham emails and spam emails [29]. However, email is currently being utilised inappropriately under the guise of Spam. Bulk or unsolicited email, generally known as spam, may contain an advertisement, a link to a phishing website, malware or a Trojan horse. Every day, each of us used to get a lot of emails, of which 70-80 percent were spam [30]. Spammers utilise spam emails to spread their spam for a variety of purposes, including hacking, phishing and banking fraud. The ideal platform for spammers to obtain user personal data and send spam emails is social media. Junk emails are another term used to describe spam. Spam emails are used to spread trojans, phishing websites, malware that looks like a virus, offers, and other types of content advertising. The term Spam stands for Self-Propelled Advertising Material [31]. Over 280 billion spam emails were sent and received worldwide in 2019. Google reports that 64% of emails sent and received in 2019 are spam emails, up from the prior years' 2%-3% rate [18]. Two different kinds of spam detection methods exist. These include spam detection based on sender and spam detection based on content [32]. Content-Type, MIME-Version, Message-ID, Return-Path, and Authentication-Results were the major elements used to detect spam sent by a specific sender [33]. When doing content-based spam filtering, the email's subject and URL are compared to the email's text to determine its text classification.

Taking advantage of the advancements in technology, a large number of cybercriminals create hazardous scam communications every day and send them to millions of individuals around the world. An easy, free, and potentially anonymous method of spreading the scams online is through email services. Spam is increasingly linked with a problematic and dangerous concern for the security, integrity, and dependability of email users on the internet, even though they typically only perceive it as annoying, uninvited advertising or a waste of time. Furthermore, spam is a significant issue because, according to estimates from Kaspersky Lab and Cisco Talos, 50-85% of the 200 billion emails received daily worldwide are spam [19]. Since spam email has been an issue for the last few decades, businesses and researchers are working to develop effective filters that are both reliable and effective. To determine if an email is spam or valid (commonly referred to as ham), various methods based on ML techniques in the literature nowadays demonstrate excellent performance with accuracies around 90% [21]. Despite the remarkable speed results and upgrades to the filters, users still report attacks and frauds originating from spam emails. The many different kinds of spam detection algorithms that have been effective in eradicating spam emails [34]. Different types of email spam detection techniques are given in figure 6:

1) CONTENT BASED EMAIL SPAM DETECTION TECHNIQUE

Emails can be automatically filtered and classified based on their contents using a variety of machine learning techniques, including k- Nearest Neighbor, Support Vector Machine, Naive Bayesian classification and Neural Networks [33], [35]. This technique often uses word analysis, occurrence analysis, and distribution analysis to detect incoming email spam by analyzing the content of the emails.

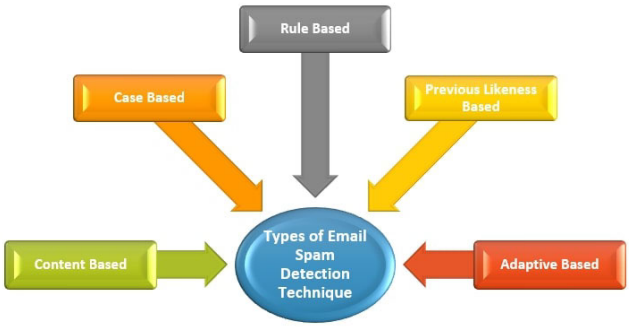

FIGURE 6. Types of email spam detection technique.

2) CASE BASED EMAIL SPAM DETECTION TECHNIQUE

Case based detection is a popular method for detecting spam emails. The first step is to gather all emails from each user's mailbox, regardless of whether they are spam or not. The next step is to pre-process the email so that it may be converted using a client interface. This involves extracting and selecting features, aggregating email data, and finally, evaluating the results. Next, the data is divided into two sets of vectors [36]. Finally, a ML method is utilised to train and evaluate datasets in order to ascertain whether incoming emails are classified as spam.

3) RULE BASED EMAIL SPAM DETECTION TECHNIQUE

In this method, numerous patterns, typically regular expressions, are evaluated against a selected message using preexisting rules or heuristics. The quality of a message improves as it acquires several matching patterns. However, the score is reduced if any of the patterns were incorrect. If the score of a message is high enough, it is classified as spam; otherwise, it is considered legitimate. Some ranking factors are static, while others need to be updated frequently to keep up with the ever-evolving threat posed by spammers and their more sophisticated and difficult-to-detect messages [37]. SpamAssassin is an excellent example of a rule-based spam detection.

4) PREVIOUS LIKENESS BASED EMAIL SPAM DETECTION TECHNIQUE

In this strategy, incoming emails are sorted according to how closely they resemble instances already stored in memory (i.e., training emails). New instances are represented as points in a multidimensional space, which is generated using the email's properties [38]. Then, the fresh instances are distributed among the most well-liked group among its k-nearest training instances. For this purpose, the k-NN method is used.

5) ADAPTIVE BASED EMAIL SPAM DETECTION TECHNIQUE

The system recognizes spam detection by assigning it to one of several categories [39]. It classifies a collection of emails into categories and assigns a defining text to each category. This technique utilizes adaptive algorithms and machine learning to remain efficient in the presence of continuously evolving spam trends.

ML methods, Fuzzy systems, and DL methods are some of the methods that have been used in email spam detection. These methods were selected due to their superior classification performance and high accuracy in detecting email spam. In the following section, ML methods are discussed in more detail. Numerous studies have demonstrated that ML delivers high performance for email spam detection across diverse datasets and evaluations metrics.

B. MACHINE LEARNING

Artificial intelligence encompasses many subfields, one of which is ML. The term ML is used to describe the process of designing, analyzing, and deploying systems that help a machine get better results. ML systems use training data to make predictions about the problem. In particular, training data is utilized to extract information and develop a method that should generalize to all conceivable problem cases throughout the learning phase [40]. ML method is used to classify new samples after learning. The goal of ML is to develop a method that predicts well on test data with new examples. For automated decision-making, ML methods are commonly used. ML uses training data to construct methods that can effectively predict fresh data outcomes, enabling automated decision-making across many disciplines [41]. Many fields have successfully used applications of ML techniques. But there's a subset that always needs new methods because of an adversarial figure, and that includes things like phishing detection spam detection and botnet identification [42]. Nonetheless, institutions and researchers need to address this issue by taking into account the unique characteristics of their respective fields of study. For instance, phishing differs from spam in that it often masquerades as legitimate-looking branch logos and requests personal information or conveys an urgent message [43]. In another study, ML security research on adversarial techniques typically focuses on spam email detection, whose adversarial figure is commonly referred to as a spammer. By including specific misspelt terms or legitimate words in the email, scammers want to fool the classifier without negatively impacting the email's readability. As a result, spam emails may contain malicious data that was purposefully injected by spammers to compromise the data used for training the classifiers and, in turn, undermine its regular operation filter [44]. As well as, a comparative analysis method in which many ML methods were tested on the same data set. Accuracy and precision were used to evaluate the various machine learning methods. The accuracy of the support vector machine is 98.09% [45]. Additionally, Cota et al. used two publicly accessible corpora. For the first set of tests, each corpus was divided into 80% training and 20% testing, and for the second set, 70% training and 30% testing. Using Random Forests, the best accuracy for the input corpus was

85.25% and 86.25 percent, respectively. These findings are consistent with other studies [46]. According to the previous study on spam detection using ML methods outlined in Table 2, it can be inferred that scholars strongly appreciate ML methods for their significance in detecting spam texts.

Currently, ML methods employed for email spam detection mostly rely on techniques such as SVM, NB, RF, and k-NN. These approaches have been successful in reaching accuracies within the range of 90-99%. Nevertheless, these strategies encounter constraints such as inaccurate positive results, unchanging feature extraction, and demanding computing intricacy. There is a notable lack of research in creating more flexible and responsive methods that can respond to evolving spam strategies. Additionally, there is a requirement to investigate hybrid or ensemble techniques that integrate various algorithms in order to enhance accuracy and minimize false positives.

ML methods have proven effective for email spam detection across multiple studies. However, ML methods may struggle with vague and ambiguous information. In contrast, fuzzy systems can better handle uncertainty and imprecision in data and logic. This is because fuzzy systems can represent and reason with vague, ambiguous information using fuzzy logic. Furthermore, fuzzy systems can adjust and adapt to changing data and situations by applying fuzzy rules. In the following section, the use of fuzzy systems is discussed in more detail in the context of email spam detection.

C. FUZZY SYSTEM

There has been a proliferation of applications of fuzzy set theory in recent years, including ML, data mining and DL. Researchers in this area recognised the need for measuring the fuzzy membership vector in a fuzzy set or event as a result of the widespread use of the idea of fuzzy set theory [55]. Additionally, Gazal et al. developed a two-level filter-based hybrid spam detection methodology. At Level-1, a high-level filter removes irrelevant and unimportant features and content. Level-2 uses a fuzzy-based composite evaluator for low-level filtration and to find the most effective features. CSDMC2010 SPAM, spambase and the SMS Spam Collection are all used in the method's implementation. The results of the comparison showed that the proposed method beat the current conventional and recent algorithms and methods, with an average accuracy of 98.80% on the CSDMC2010 dataset, 97.79% on the spambase dataset, and 98.84% on the SMS Spam collecting dataset [56].Moreover, fuzzy inference systems utilising Interval Type-1 and Interval Type-2 were created employing four distinct machine learning algorithms to showcase their efficacy in identifying spam. The methods evaluated were SVM, LR, and average perception. The Interval Type-2 Mamdani fuzzy inference system (IT2M-FIS) demonstrated superior performance, with an accuracy of 0.955, recall of 0.967, F-score of 0.962, and area AUC of 0.971 [57]. Another aspect, Srinivasarao et al. introduced fuzzy-based Recurrent Neural network-based Harris Hawk optimization (FRNN-HHO) to post-classify spam and ham messages. Three distinct datasets SMS, Email and Spam-assassin are used to assess the efficacy of the proposed architecture. For the SMS dataset, the suggested method achieved an AUC of 0.9699, for the email dataset it achieved 0.958, and for spam assassin it achieved 0.95 [58]. In another study, fuzzy C-Means clustering was utilized for spam email segmentation to prevent cybercrime in the Internet era. Previous studies have shown that clustering in data mining for spam filtering has been understudied. This study demonstrated that Fuzzy C-Means clustering showed promising results for spam email categorization on a public spam dataset using different parameters [59]. As well as email's growing popularity as a secure online communication method has led to the rise of unsolicited bulk emails or spam. A proposed spam filtering strategy handles this issue by employing relief feature selection and a fuzzy-SVM to deal with uncertain elements. Experiments showed that these algorithms improved spam filtering accuracy and detection speed [60]. In another study, the widespread problem of spam in mailboxes has negative effects on network resources and daily life. To address this issue, a content-based spam filtering algorithm using fuzzy- SVM, and k-means was proposed. k-means clustering reduces data while maintaining critical information. Meanwhile, fuzzy-SVM trains a classification method to handle ambiguity. This strategy improves spam filtering speed and accuracy, according to experiments [61]. Table 3 presents prior research on spam detection using fuzzy system. From this analysis, it can be assumed that researchers highly value the significance of fuzzy system techniques in email spam detection.

The present research examines the application of various fuzzy systems in email spam detection. It focuses on distinct models, datasets, merits, and findings. However, there is a significant lack of research in combining fuzzy logic with sophisticated DL methods. Although Fuzzy-BERT demonstrates potential, there is a lack of investigation into hybrid models that integrate fuzzy logic with other cutting-edge algorithms in order to enhance accuracy and resilience. Moreover, the majority of research primarily concentrate on binary classification, disregarding the potential advantages of employing multi-class classification methods for spam detection.

Fuzzy systems have proven effective for email spam detection across multiple studies. Fuzzy systems provide advantages in dealing with uncertainty but require expertise in design and may struggle with high-dimensional data. In contrast, DL methods can handle high-dimensional data. DL can automatically learn complex patterns from raw text input without extensive feature engineering. This enables DL methods to overcome the curse of dimensionality faced by fuzzy systems in processing raw email data. DL methods can learn directly from raw text while handling high dimensionality [63]. In the following section, the use of DL

methods is explored further for email spam detection, as DL is well-suited to overcome limitations of fuzzy systems.

D. DEEP LEARNING

DL is an up-and-coming field that uses several nonlinear processing layers to learn features directly from the input, leveraging AI and ML. Email spam detection accuracy may be greatly improved with the help of DL methods. Deng and Yu conducted an analysis of different DL methods, categorising them into supervised, unsupervised, and hybrid deep networks based on their network structures. They also explored various applications of these techniques, including computer vision, language modelling, text processing, multimodal learning, and information retrieval [64], [65]. DL relies on representations of data that include several levels of hierarchy, often in the form of a neural network with more than two layers. Data features from a higher level can be spontaneously integrated into those from a lower level using these methods. Each neuron in a neural network (NN) shares several common characteristics. The number of neurons and their interconnections are in turn determined by the nature of the application being used [66]. Another aspect, Baccouche et al. introduced a multi-label LSTM model to identify spam and fraud in emails and social media posts. The model was developed by merging two datasets. The system was trained by utilising a collective dataset of prevalent bigrams obtained from multiple sources. Their model has an accuracy of 92.7%. A limitation of the study was the absence of a comparative analysis with other sophisticated techniques for identifying harmful information. In the future, they intend to explore alternative NLP methods in order to enhance the accuracy of the model [67]. In this study,Alauthman et al. proposed the utilisation of a SVM andGRU-RNN approach to detect botnet spam emails. Engaging with a dataset containing spam records. According to their assertion, their method attained a precision of 98.7%. Their research was limited to assessing the efficacy of the proposed model using a single dataset. The proposed method accurately identifies spam emails, but additional investigation is required to enhance the GRU model by integrating supplementary multiclass classifiers [68]. Moreover, AbdulNabi and Yaseen et al. conducted research on word embedding techniques for the purpose of classifying spam emails. The scientists enhanced the performance of a pre-trained BERT model and conducted a comparison with DNN and traditional classifiers such as naïve Bayes and k-NN. The proposed technique attained a 98.67% accuracy and a 98.66% F1 score when evaluated on two open-source datasets [69]. Furthermore, Eckhardt and Bagui et al. designed a study in which they analysed LSTM and CNN methods for the purpose of classifying textual input. The investigation revealed that the LSTM method achieved the maximum accuracy of 98.32% and a ROC score of 96.57%. The comparison just pertains to the classification of textual material. They asserted that the Adam optimizer outperformed the SGD optimizer in both models. According to the study, ReLU demonstrated superior performance compared to CNN, while sigmoid showed superior performance compared to LSTM on average [70]. As well as Rafat et al. investigated the impact of text pre-processing on email classification using ML and DL techniques. The ML and DL algorithms were compared using the Spamassassin corpus, both with and without text pre-processing. The researchers discovered that DL methods performed better than ML methods. Specifically, the LSTM method achieved a precision of 95.26%, recall of 97.18%, and an F1-score of 96% without any text pre-processing. [71]. Additionally, Wen, Tingke, et al. introduced LBPS, a phishing scam detection model for blockchain financial security. The model is built on LSTM-FCN and BP NN. The proposed model utilises a Backpropagation Neural Network (BP NN) to analyse implicit features and a LSTM-FCN NN to analyse the temporal aspects of transaction data. The experimental findings, using Ethereum data, demonstrated that the chosen characteristics effectively identified fraudulent accounts involved in phishing scams, achieving a 97.86% F1-score and a 97% accuracy rate [72]. Table 4 presents the previous research on spam detection using DL methods. DL methods undoubtedly enhance the effectiveness of the spam detection method, reduce the impact of overfitting, and handle large data.

A comprehensive explanation of the many different DL methods that can be used to detect spam in email, including models such as CNN, LSTM, and hybrid combinations of these methods. There is a significant research gap in the development of ensemble learning techniques, which combine the strengths of many DL models to further boost performance. This is despite the fact that the results have been promising. In addition, although a great number of studies make use of datasets that are accessible to the public, there is a dearth of research that investigates the application of these models to large-scale datasets that are based on the actual world and have the potential to more accurately represent a variety of spam characteristics. There is also a lack of attention paid to the interpretability and explainability of DL models, which are essential for the actual implementation of spam detection systems. This is another gap. In addition, the majority of the research that is currently being conducted place an emphasis on accuracy measures, while ignoring other significant features like as processing efficiency and adaptation to increasingly sophisticated spam strategies. By addressing these deficiencies, it may be possible to develop spam detection systems that are more robust, efficient, and adaptable through the application of DL techniques.

The present review diverged from the previous reviews by placing greater emphasis on reevaluating ML, fuzzy system, and DL methods employed for the purpose of detecting email spam. The review aims to discuss email spam detection methods, the parameters utilized for comparative analysis, simulation tools, and the dataset corpus. The reviewed era encompasses recent research articles that contribute to

TABLE 2. ML methods for email spam.

Reference	Dataset	Merit	Methods	Limitation	Results
[33]	CSDMC2010	Proposes an algorithm that aims to enhance the effectiveness of spam detection by mitigating redundancy and conflicts across criteria.	k-NN, SVM, RF, NB, DT and Adaboast,	The analysis fails to investigate the effects of various feature selection methods between feature complexity and computational performance.	Accuracy: 95.97 % Precision: 95.63% Recall: 95.41% F1-score: 95.52%
[47]	Enron, CSDMC2010, and TurkishEmail	Focuses on the issue of effectively managing linguistic variations and complexity within the context of spam filtering.	SVM, LR and NB	The lack of a discussion about the potential limits or drawbacks of the artificial bee colony algorithm employed in the proposed method.	Accuracy: 98.91 %
[48]	The dataset obtained from Kaggle consists of 4,360 non-spam samples and 1,368 spam samples.	Provides a comparative comparison of various ML techniques.	SVM, k-NN, LR, NB, RF and DT.	It does not provide information about the exact dataset that was used to train and test the spam detection.	Accuracy: 99% Precision: 97% Recall: 99% F1-score: 98%
[49]	The dataset used in this paper is retrieved from Kaggle and contains 5171 emails, with around 71 of them being ham and 29 of them being spam.	Covers Content Based testing as the main metric and the importance of email content in spam classification.	SVM, NB and RF.	The impact of different feature extraction techniques is not explored on the performance of the algorithms.	Accuracy: 98.41% Precision: 100% Recall: 100% F1-score: 98%
[45]	Spambase	Provides a comparative analysis of different methods.	SVM, NB, DT, LR and k-NN	Does not introduce any novel techniques for spam detection.	Accuracy: 98.09% Precision: 96.39%
[50]	Lingspam	Proposes modified PMI-based feature selection method.	k-NN	Does not explore the performance of the K-NN classifier with other distance metrics	Accuracy: 98.06% F1-score: 95.98%
[51]	LingSpam	Proposes a score based SVM method for spam email detection.	SVM, k-NN	No discussion of the efficiency of the proposed method in handling large volumes of email data or real-time spam detection.	Accuracy: 98% Precision: 97%
[52]	Spamassassin	Employs a list of the most prevalent spam characteristics to raise the rate of spam detection	MLP, NB, RF, and DT	Utilize a limited set of features derived from the corpus.	Accuracy: 99% Precision: 99.60% Recall: 96.30%

TABLE 2. *(Continued.)* ML methods for email spam.

[53]	Enron dataset	Aims to achieve better accuracy in the detection process compared to previous methods.	ELM and SVM	Does not provide a detailed discussion of the specific features used in the ELM and SVM methods.	Accuracy: 94.06%
[54]	Weibo social network data	Uses features of text and behavior to find spammers.	SVM	Proposed solution relies on a labeled dataset that was manually classified, which may introduce bias and limit the generalizability of the results	Accuracy: 99.5% Precision: 99.90% Recall: 99% F1-score: 99.7%

the progress of email spam detection systems. Different email spam detection methods exhibit varying strengths and weaknesses, influenced by factors such as dataset size and complexity. An analysis of the most effective techniques along with their internal workflows is provided in the following section.

III. METHODS

Email spam refers to the sending of fraudulent or undesired mass emails through either an individual's account or an automated mechanism. The prevalence of spam emails has steadily risen over the past decade, posing a widespread issue. ML and DL have significantly contributed to the identification of spam emails. Researchers are utilizing a range of methods and strategies to create innovative spam detection. In This section will provide an overview of the most widely used ML and DL methods that have been optimized for spam detection.

A. SUPPORT VECTOR MACHINE

The SVM is a supervised learning paradigm with connected learning method used for categorization of input data. Any information fed into a computer that may be represented by a vector representation is considered input data. SVM's great accuracy and precision in classifying different classes of data have led to its widespread adoption [82]. It specializes in unstructured data, making it suitable for classifying both linear and non linear datasets. Non-linear (SVMs) are used to categories data received by a computing device, while linear SVMs are helpful only for certain types of data. Its benefits include its efficiency in high-dimensional settings and its adaptability [83]. The downside of this approach, however, may be the lack of transparency in the output, which makes it hard to evaluate the results [84]. Figure 7 presents structure of SVM:

A novel approach to identifying spam in electronic messages. That was accomplished through the use of Naïve Bayes or SVM based Supervised ML. They tested various algorithms to see which ones were best at identifying spam from regular correspondence. NB accuracy was 95%, whereas the first method based on SVM achieved an

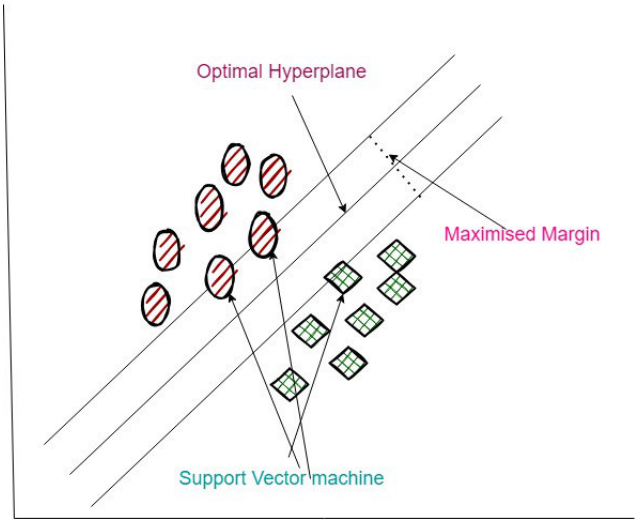

FIGURE 7. Structure of the SVM.

impressive 97.5% [85]. Additionally, a comparative analysis method in which many ML methods were tested on the same data set. Accuracy and precision were used to evaluate the various machine learning models. The accuracy of the support vector machine is 98.09% [45]. Furthermore, a Comparative studied SVM, Random Forest and Multimodal NB are the three methods of content-based e-mail spam detection. The advantages and disadvantages of the three approaches were compared in terms of their usefulness and effectiveness. The results of the experiment showed that SVM perform the best, with the other methods trailing behind by only a very small margin [33]. Moreover, a new online method used binary representation and linear SVMs without feature selection. Character n-gram models allow the authors to predict all features. The next strategy showed more features and yielded millions of unique 4-gram features from TREC corpora [86]. As well as, It's crucial to recognise that most spam and valid messages use a template. SVM based incremental clustering algorithm was used by Haider in 2007 to identify spam and non-spam email messages based on their contents [87]. Discerning the importance of fine-tuning

TABLE 3. Fuzzy system for email spam.

Reference	Dataset	Merit	Methods	Limitation	Results
[62]	Enron, PU, Lingspam	Added a fuzzy-based composite evaluator at level-2 for low-level filtration and identifying the best features.	Fuzzy-BERT.	Does not discuss spam classification false positives and negatives and their effects.	Accuracy: 98 % Precision: 99% Recall: 99%
[56]	CSDMC2010 SPAM, Spambase, and SMS Spam Collection	Detailed investigation of feature filters and selection strategies to improve spam message detection performance and reliability	Fuzzy-RF and NB.	Not addressed are the method's scalability, computational difficulty, and generalizability to different datasets or languages.	Accuracy: 98.80 % Precision: 95% Recall: 97%
[61]	Spambase	Quadratic programming is used to solve the fuzzy - SVM optimal classification hyperplane problem.	Fuzzy-SVM and k-means.	lacks discussion of parameter choices and algorithm sensitivity.	Accuracy: 90% Recall: 93%
[60]	Spambase	Presents a novel approach for feature selection utilizing the relief feature selection technique	Fuzzy-SVM	Not compares the proposed spam filtering method regarding others.	Precision: 94% Recall: 92.59%
[57]	Twitter dataset	Fuzzy-LR, SVM and BPM.	Evaluates classification methods and shows the possibilities of fuzzy logic-based social network for spam detection.	The study focuses on spam detection in social networks, notably Twitter, and does not examine the application of the presented methodologies to other domains.	Accuracy: 95.50 % Recall: 96.70% F-score: 96.20%

classification algorithms. Optimize SVM algorithm for email spam detection is shown in algorithm 1.

B. DECISION TREE

The DT is a popular technique for classifying data since the solution it produces is both interpretable and straightforward. Furthermore, it provides a result more quickly than other categorization techniques [88]. It is structured like a tree with a central hub, branches, and leaves. The terminal node, or leaf node, represents a class attribute, and the other nodes represent potential solutions. To determine the class properties of the terminal node, the route from the root to the terminal node must be accurately traced [89]. Tracing the tuples will be made significantly simpler by the translation of the tree into categorization rules [90]. It was common practice in the fields of data mining, machine learning, and even statistics to employ the decision tree learning method. Spam detection has been modified to use DT learning. The structure of the DT is presented in figure 8.

A hybrid approach combining LR and DT is used for email spam identification. LR was employed to reduce the impact of noisy data or instances prior to supplying the data to DT induction. By applying a predetermined false-negative threshold, LR effectively eliminated the noisy data by selecting only the accurate predictions [91]. This study used Spambase dataset to assess the proposed technique. 91.67% accuracy is encouraging for the given strategy. LR may increase DT performance by minimising noisy data. GADT is a hybrid spam email detection method. PCA improved GADT's performance. Decision tree

TABLE 4. DL methods for email spam.

Reference	Datasets	Merit	Methods	Limitations	Results
[69]	Spambase	The utilization of BERT, a word embedding method, enhances the efficacy of spam identification.	BERT	Require an extensive input sequence to enhance the training of the method.	Accuracy: 98% F1-score: 98%
[73]	Twitter social Honeypot, Twitter 1KS-10KN	Identifies spam content by efficiently utilizing minimal computational resources.	CNN	The complicated structure of the method.	Accuracy-91.36%
[74]	WEBSPAM-2007	The cognitive capability enables the search engine to automatically detect webspam.	LSTM	Require optimization of the algorithm to effectively process vast volumes of data obtained from the internet.	Accuracy: 96.96% F1-score: 94.89%
[75]	Spanassian	Offered instructions for enhancing offline data.	NN	Required to augment the offline dataset in order to improve the performance of the method.	Accuracy: 99.07%
[76]	800 Turkish emails dataset	Proposed hybrid Method.	ANN, LSTM and Bi-LSTM	Small dataset	Accuracy: 100%
[77]	Enron corpus	Integrates a CNN, LSTM to create a multi-modal architecture based on method fusion (MMA-MF) for spam filtering. This architecture successfully removes spam that is hidden in text or images.	CNN, LSTM	Does not compare the suggested MMA-MF method to other state-of-the-art spam filtering algorithms, making it difficult to evaluate its performance.	Accuracy: 98.46% Precision: 98.30% Recall: 98.52% F1-score: 98.45%
[78]	SMS Spam	Experimental data illustrate the LSTM outperforms earlier spam detection methods.	LSTM	Lack of information about preprocessing methods like eliminating special symbols and converting to lowercase.	Accuracy: 98.50%
[79]	SMS Spam	The proposed method uses fewer training parameters and takes less time than current DL classifiers.	LSTM, Lightweight-GRU	Does not analyze how WordNet improves SMS text input interpretation.	Accuracy: 99.04%
[80]	Spambase, Ling spam	Discusses LSTM hyperparameter determination.	LSTM	No discussion of the efficiency of the proposed method in handling large volumes of email data or real-time spam detection.	Accuracy: 97.4%
[81]	Self-generated emails dataset	The utilization of word embedding in CNN yields the highest level of accuracy.	CNN and LSTM	Tested English corpus only.	Accuracy: 96.34%

Algorithm 1 SVM Algorithm for Email Spam Detection

Input: Email message x to classify
Input: Training set S, kernel function k, regularization parameters $C = \{c_1, \ldots, c_{\text{num}}\}$, kernel coefficients $\gamma = \{\gamma_1, \ldots, \gamma_{\text{num}}\}$
Input: Number of nearest neighbors for k
for $l = 1$ to num **do**
  Set $C = c_i$
  for $j = 1$ to num **do**
    Set $\gamma = \gamma_j$
    Train SVM classifier $f(x)$ with parameters (C, γ) on S
    if first classifier **then**
      Set $f_{(x)} = f(x)$ as best classifier
    else
      Compare $f(x)$ with $f_{(x)}$ using k-fold cross-validation
      Set $f_{(x)}$ to the more accurate classifier
    end if
  end for
end for
Return spam or ham classification of x using final classifier $f_{(x)}$

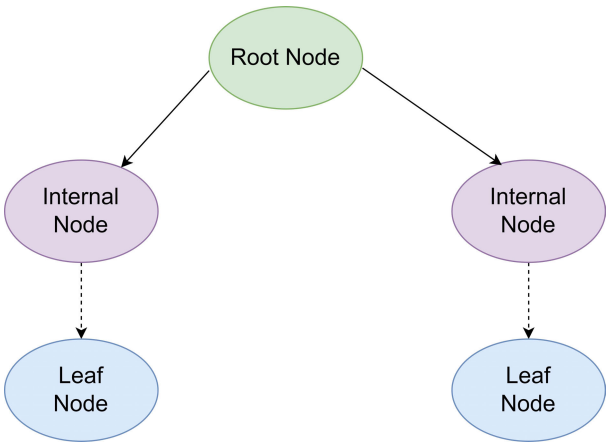

FIGURE 8. Structure of the DT.

classification models with and without Recursive Feature Elimination (RFE) were investigated for spam detection [92]. Furthermore, a novel spam detection system that reduced false positives by mislabeling nonspam as spam. First, wrapper-based feature selection. Second, C4.5 was used to train the decision tree classifier model. Third, the cost matrix weighted false positive and false negative errors differently. The MBPSO-selected decision tree had 91.02% sensitivity, 97.51% specificity, and 94.27% accuracy [93]. Moreover, the suggested method combines particle swarm optimization with unsupervised filtering to enhance accuracy to 98.3%. Comparative analyses indicate better results than current methods [94]. The optimize DT algorithm for email spam detection is shown in algorithm 2.

Algorithm 2 DT Algorithm for Email Spam Detection

Input: Email message dataset D
Calculate entropy $H(D)$ of full dataset
while stopping condition not met **do**
  for each attribute A **do**
    Calculate entropy $H(D|A)$ for splits on A
    Calculate average entropy over all splits
    Calculate information gain $Gain(A)$
  end for
  Choose A with highest $Gain(A)$ as split attribute
end while
Return DT method classifying messages as spam or ham

C. K-NEAREST NEIGHBOR

The k-Nearest Neighbor (k-NN) algorithm is one of the most popular since it is simple to use and understand. This is because its advanced features can be quickly grasped and put to use [95]. k-NN uses the computed distance between a given instance and its k-NN to determine how to categorize the instance in question. To which category a dataset belongs is decided by how many votes are cast for each possible nearest neighbor value. If k is set to two, for example, the dataset will be classified based on its distance from its two nearest neighbors [96]. The Euclidean distance (ED) between a specified training sample and a test sample is typically used for this purpose [97]. The classification results for k-NN vary greatly depending on the value of k chosen for the number of neighbors. A simple k-NN structure is given in figure 9.

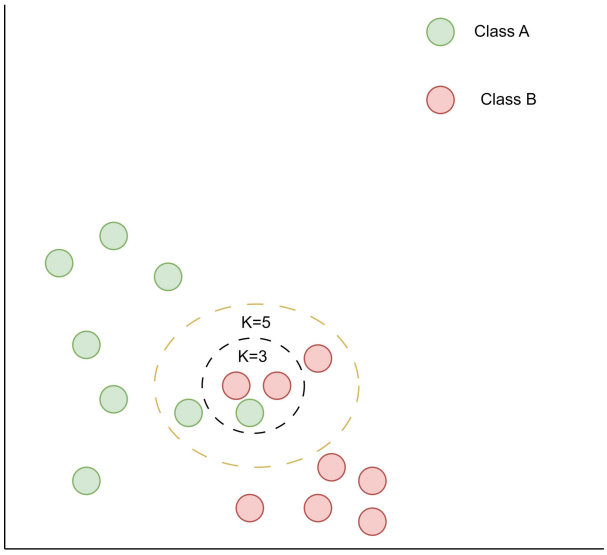

FIGURE 9. Structure of the k-NN.

Using the k-NN classifier, Sahin et al. developed a filtering approach to pick features for spam detection via email [98]. Another aspect, the experiments calculate the accuracy and F-measure of the e-mail texts classification using various feature selection methods, varying numbers of features, and two distance measures to determine how far apart examples

in the dataset were when executing the k-NN classifier. The percentages of success gained were 98.08% and 95.98%. They suggested an approach that combines SVM and k-NN. The determination approach they came up with uses names and proximity to a restriction on choices to determine which instances to pick. The basic idea was to find similar questions and construct a neighboring SVM that jelly the separation process on the set of similar questions [99]. Furthermore, they conducted experiments using the publicly available dataset Dredze, which demonstrated an improvement in accuracy of almost 98%. In order to combat spam, they employed k-NN text classification using Chi squared feature selection to filter out unwanted messages. The value of K where the k-NN classifier obtains the highest accuracy was found through experimentation [50]. Hnini et al. proposed using three Nearest Neighbour (NN) methods k-NN, Wk NN, and K-d tree to detect spam. NLP pre-processes emails and extracts features using Bag-of-words (BoW), N gram, and TF-IDF. k-NN performed well on four measurement parameters in Enron and LingSpam datasets [100]. Additionally, a new spam categorization method that combines the Harris Hawks optimizer (HHO) and k-NN algorithms. This study found that the proposed spam detection method had the highest classification accuracy. The proposed approach achieved 94.3% accuracy in experiments [101]. The k-NN method for email spam detection is presented in an algorithm 3.

Algorithm 3 k-NN Algorithm for Email Spam Detection

```
Extract class labels (spam or valid) for each email in the training and test datasets
Set k = number of nearest neighbors
Load test set emails into D
Load training set emails into T
Initialize empty label set L to store classifications for test emails
Load training data
Load test data
for each test email d and each training email t do
    Initialize empty set Neighbors(d) for nearest neighbors
    if number of neighbors < k then
        Find k closest matches to d from T and add to Neighbors(d)
    end if
    if number of neighbors ≥ k then
        Classify test email d based on labels of k nearest neighbors in T
    end if
end for
Return final classifications of emails in test set D as spam or ham
```

D. RANDOM FOREST

The RF is a prime example of an ensemble learning strategy and regression method well suited to the solution of issues involving the categorization of data. Tin K. Ho first presented the generic random forest in 1995, then in 2001, an expansion of this approach. There are a lot of decision trees in this method. Rather than creating each tree using the same set of features, it generates a random forest of trees whose collective prediction is more accurate than that of any one tree [102]. The approach relies on the fact that creating a simple decision tree with a limited number of features requires nothing in the way of processing resources [103]. The algorithm's three primary hyperparameters are node size, tree depth, and feature sampling. A simple RF structure is given in figure 10.

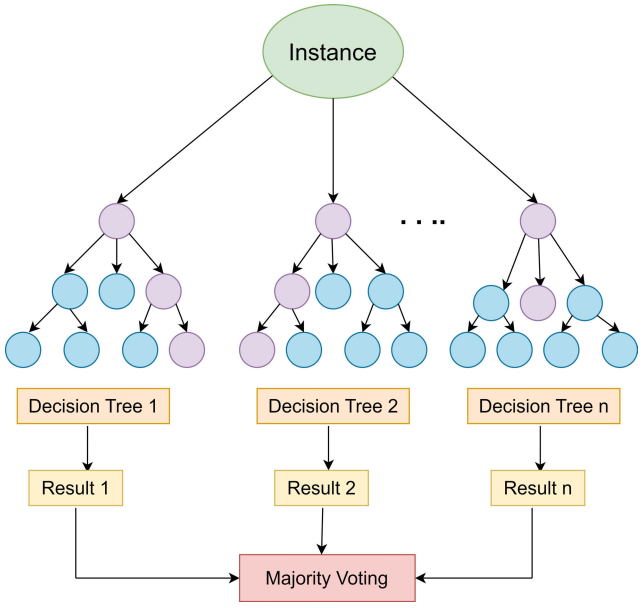

FIGURE 10. Structure of the RF.

The study of many Spam Filtering tactics and the discussion of spam message categorization using various Machine Learning algorithms for the Spambase dataset are brought to a close by the methods proposed for multiple parameters. RF has a higher accuracy 94.87% than other Machine Learning techniques [104]. Furthermore, Cota et al. used two publicly accessible corpora. In the initial set of tests, the corpus was split into 80% for training and 20% for testing. In the second set, the split was 70% for training and 30% for testing. Using RF, the best accuracy for the input corpus was 85.25% and 86.25 %, respectively. These findings are consistent with other studies [46]. On the other hand, Shrivastava et al.'s Weka implementation makes use of cross-validation and a training set. For the training set, it's going to be the same data for both purposes. The training set is additionally split up into many folds for the purpose of cross validation. As a result of this implementation and experimentation, it's been to the conclusion that using a training set with a Random Tree classifier yields approximately 100% accuracy in just 0.01 seconds [105]. Moreover, Goh et al. improved performance by boosting, bagging, rotating forest,

and stacking. SVM's high accuracy would be substantially harmed by tainted datasets, hence the authors suggest MLP. The algorithm with the highest AUC was RF with AdaBoost, at 93.7% [106]. Therefore, Random Tree is the most effective technique for identifying spam e-mail. The RF algorithm for email spam detection is shown in algorithm 4.

Algorithm 4 RF Algorithm for Email Spam Detection

Input: X: number of nodes per decision tree
Input: N: number of features per email message
Input: Y: number of decision trees to train
while termination conditions not met **do**
    Randomly select email message S from training data
    Grow decision tree R_t from S, with maximum depth X
    Randomly select n features to split on at each node, $n \ll N$
    Compute optimal split point over the n features
    Split node into two child nodes based on optimal split
end while
Add decision tree R_t to forest
repeat
    Repeat steps 5-9 until maximum number of nodes X reached
until Y times to grow forest of Y trees
for new email message **do**
    Pass down each of the Y trees to reach a leaf node
    Classify email as spam or not based on leaf node majority vote
end for
Return final classification spam or ham

E. NAÏVE BAYES

The NB classification is both a supervised learning method and a statistical approach to classification. It serves as an important probabilistic method and allows us to exploit ethical grey areas by manipulating the odds of the method's predictions. Analytical and prescriptive issues can be solved with its help. Thomas Bayes created the categorization method now known as Bayesian analysis [107]. Useful learning algorithms are provided by the categorization, and both historical information and new experimental data can be combined. In order to better understand and evaluate various learning algorithms, NB Classification provides a useful perspective [108]. The algorithm is robust to noise in the input data and is capable of accurately calculating likelihoods for hypotheses. A simple NB structure is given in figure 11.

Sahami first proposed the NB algorithm for spam detection, and it has since found widespread use in commercial spam filters and open-source spam classifier implementations thanks to its high accuracy in conducting binary classification and straightforward implementation. The researchers initially applied the NB method to the spam filtration problem in

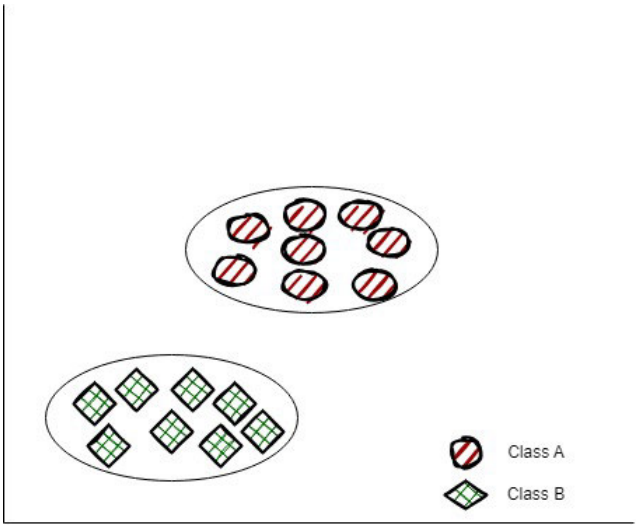

FIGURE 11. Structure of the NB.

1998, where the problem of message classification was examined within a statistical decision theory framework [109]. Another aspect, the Bayesian framework is superior to other algorithms because it integrates evidence from a variety of sources. By employing information gain to select binary features, the researchers conducted experiments on two private corpora. The results of these experiments showed that the inclusion of non-textual elements enhanced the classifier's capability to categorise the messages. This finding provides support for the NB filter's capacity to maintain low false positive rates. A novel method wherein incoming emails' URLs (links) were categorised using a NB model [[110]. Furthermore, this filter used delayed feedback to be periodically refreshed with all of the messages that had been classified by the system, not just the ones that had been erroneously classified. Both spam and non-spam emails with at least one URL were included in the experiment's private corpus. It was determined that the system provided the same level of performance as other URL or keyword based filters, with the exception that this model did not necessitate maintaining a blocklist or white list corresponding to the URLs, making it fully automated [111]. Moreover, Ciltik et al. designed and evaluated the approaches under two models: a class general model and an email specific model. When the two models are integrated, the latter is used in situations where the former fails. However, the proposed system and the techniques created are universal and can be used with any language. Extensive testing was conducted, and results showed a success rate of roughly 98% for Turkish and 99% for English. Time complexity has been demonstrated to be greatly decreased without impacting performance [112].As well as, spam emails' influence on privacy and productivity. They use NB, SVM and RF classifier to screen spam emails. NB algorithms reliably recognise and classify spam and unwanted emails, with accuracy rates up to 98.8% [113]. The

algorithm for email spam detection using NB is presented in algorithm 5.

Algorithm 5 NB Algorithm for Email Spam Detection

```
Input email message dataset
for each email message do
    Split message into individual component tokens
    for each token do
        Calculate spam probability S[W] = C_spam(W) / (C_spam(W)+C_ham(W))
        Store spam probability values in database
    end for
end for
for each email message M do
    Initialize spam score I[M] = 0
    while not end of message do
        Get next token T_i
        Query database for spam probability S(T_i)
        Update message's spam probability S[M]
        Update message's ham probability H[M]
        Compute message filtering signal:
        I[M] = I[M] + S[M] − H[M]
        if I[M] > threshold then
            Label message as spam
        else
            Label message as ham
        end if
    end while
end for
Return final email classification as spam or ham
```

F. ARTIFICIAL NEURAL NETWORK

ANN is a computational approach that draws inspiration from the structure and functioning of biological neural networks, such as the human brain. An ANN is composed of interconnected artificial neurons organised in layers. The input neurons receive data, the hidden neurons process information, and the output neurons generate results [114]. The power of ANN stems from the connections between these neurons which have adjustable weights that are tuned during training [115]. By dynamically adapting the weights to match input and output values from the training data, ANN can approximate the mapping function representing relationships in the data. Information flows through the network hierarchy starting from the input layer [116]. Each neuron's activation is determined by the input data, connection weights, and the activation function, which manages how inputs are transformed [117]. A simple architecture of the ANN is presented in figure 12

Zhan and his team conducted research on spam classification using the ANN method. Their approach utilises descriptive qualities of the evasive patterns employed by spammers, rather than relying on the context or frequency of terms in the message. Over several months, the researchers compiled

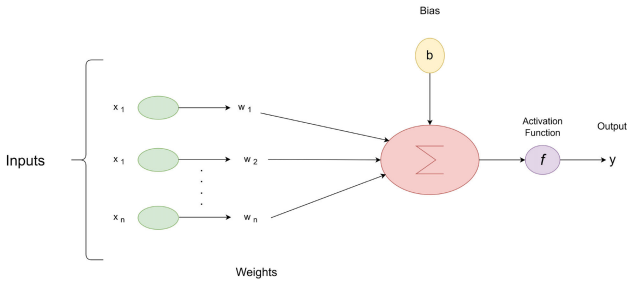

FIGURE 12. Structure of the ANN.

a dataset consisting of 2788 legitimate and 1812 spam emails to train and evaluate their model [118]. Additionally, spam email detection models challenges, as it wastes Internet traffic and enables phishing and malware attacks. To address this, a feature selection-based strategy employing the sine-cosine algorithm (SCA) to optimize ANN for spam detection is proposed. Experiments showed the suggested ANN classifier surpassed other methods, achieving accuracy, precision, and sensitivity of 97.92%, 98.64%, and 98.36%, respectively [119]. In this study, an ANN that has been tuned using the Grasshopper Optimization Algorithm (GOA) is used to create a new method for email spam identification. The suggested GOA-ANN method outperforms traditional methods in experiments, achieving 94.25% accuracy in classifying spam. The research shows how bio-inspired algorithms, like GOA, can be used to improve ANN learning for better spam detection [120]. Furthermore, the challenges of constructing efficient ANN structures and tuning parameters for spam detection are examined. A hybrid model combining a genetic algorithm (GA) with an ANN is proposed to optimize spam detection capabilities. Experiments showed the hybrid ANN-GA model performs better in spam detection than conventional ANN methods [121]. Despite taking longer to train, neural networking can classify new patterns and tolerate noisy data. The algorithm for email spam detection using ANN is presented in algorithm 6.

G. CONVOLUTIONAL NEURAL NETWORK

As a type of DL method, CNN has recently risen to prominence in the field of computer vision and is gaining attention in other areas, such as defending against email spam. In recent years, CNNs have been a popular topic of study. CNN is useful because it can handle errors well, process information in parallel, and learn on its own. It has been used in the area of email spam filtering with great success. CNNs were described by Albelwi as a type of DL that is based on biology [122]. The network's neurons have weak local connections and a relatively even weight distribution.A CNN is constructed by stacking multiple trainable layers on top of each other. This is then followed by a supervised classifier and a collection of arrays known as feature maps, which represent the input and output of each layer. A typical CNN consists of several layers, such as a

Algorithm 6 ANN Algorithm for Email Spam Detection

Input sample email message dataset
Initialize method parameters $\mathbf{w}$ (weight vector) and b (bias term) randomly or to 0
repeat
  Get a training message sample $(\mathbf{x}, c)$ that our current method misclassifies, i.e. sign$(\mathbf{w}^T\mathbf{x} + b) \neq c$
  if no such misclassified sample exists **then**
    Training completed, store final $\mathbf{w}$ and b and stop
  else
    Update parameters:
    $\mathbf{w} = \mathbf{w} + c \cdot \mathbf{x}$
    $b = b + c$
    Go to step 1
  end if
until
To classify new email message $\mathbf{x}$:
Compute sign$(\mathbf{w}^T\mathbf{x} + b)$
Return email message classification (spam or ham)

convolutional layer, a pooling layer, and a fully connected layer. The utilisation of multiple layers in CNN enables the automatic acquisition of feature representations that are extremely distinguishable, eliminating the need for manually engineered features [123]. A conventional backpropagation neural network (BPN) operates on individual manually created image data, but a CNN is designed to extract valuable and essential characteristics from an email in order to classify it. A simple architecture of the CNN is presented in figure 13.

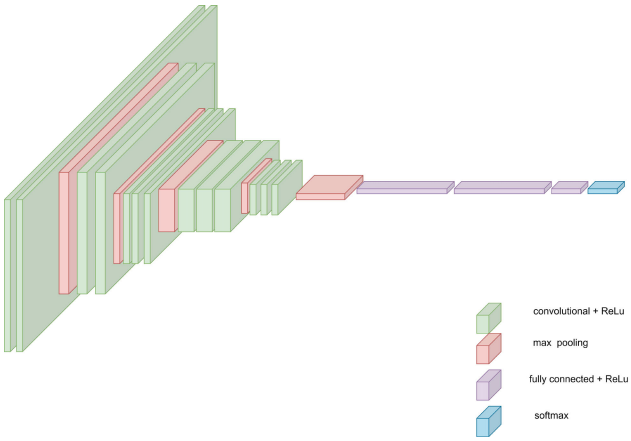

FIGURE 13. Structure of the CNN.

A compared SMS detection using DL classifiers, AI, and CNN have been performed by [124]. CNN achieved the best accuracy of 99.10% and 98.25% on SMS Spam Collection v.1 and Spam SMS Dataset 2011 12, respectively. Another aspect, the SMS Spam Collection dataset categorizes spam and ham text messages using CNN and Long Short-term Memory (LSTM). CNN and LSTM models extracted and categorized vectors. Three CNN layers with dropouts yielded 99.44% accuracy [20]. Moreover, Gupta et al. studied the efficacy of eight different classifiers and compared their results. The results of the classifier evaluation show that the CNN classifier achieves a maximum precision of 99.19% and an Average Recall of 99.26% and 99.94% respectively, across the two datasets [125]. As well as a CNN method was developed for SMS spam detection using the Tiago dataset. After preprocessing the text data, including tokenization and stopwords removal, the CNN achieved 98.40% accuracy in classifying messages as spam or not spam. The work provides a highly accurate CNN architecture and process for SMS spam detection [126]. In another study, the analyses images using CNN and compares the findings to other ML methods. The CNN-based methodology detects real-world image spam and challenging image spam-like datasets better than earlier methods by using a new feature set mixing raw photos and Canny edges [127]. The algorithm for email spam detection using CNN is presented in algorithm 7.

Algorithm 7 CNN Algorithm for Email Spam Detection

Input Email Message
Input parameters
file getting ()
label `getlabel`(file)
test `gettest`(file)
vec `getword2vec`()
random (label)
while condition **do**
  `Nf_CV`(len (`Xshuffle`), nf)
  for `trindex`, `teindex` in kf **do**
    `Xtotal, vtotal` ← `xshuffle[trindex], yshuffle[trindex]`
    `Xtrain, Xdev, vtrain, vdev` ← split (`Xtotal, vtotal`)
    for $j < N$ **do**
      get conv ()
      $h \leftarrow$ `sigmoid`(conv)
      $N \leftarrow$ `getk`()
      `tensorr` ← `gettensor`()
      for `X_v` in `Xtrain, vtrain` **do**
        value, `indice` ← `topk` (`tensorr`)
        `tensors_get` (value, `indice`)
        `tensora_append` (tensors)
      end for
    end for
    `con` (`tensorp`)
    `con_sigmoid` (`con`)
    get softmax (`conn`)
    if getdev() **then**
      tr ← false
    end if
  end for
end while
return Final Email Message Detection (Spam or Ham)

H. LONG SHORT-TERM MEMORY

LSTM is an advanced RNN in sequence modeling. RNNs function work in a similar way the network remembers earlier information and utilizes it to process the current input [128]. RNNs with traditional architectures have a recurring problem. Because of the phenomenon known as the vanishing gradient, RNN) are incapable of retaining and recalling long-term dependencies. LSTM is specifically designed to mitigate risks related to long-term reliance [129]. The default behavior of LSTM is to learn long-term dependencies by memorizing information over lengthy periods of time. LSTM employs gates to regulate information flow in recurrent computations. LSTM was designed in 1997, this type of recurrent neural network to deal with temporal data sequences and to solve the challenges of expanding and vanishing gradients, which is a problem [130]. A memory cell is included in this neural network which can hold values that have been recorded throughout time in relation to previous information. The memory cell is controlled by three gates. Each of the gates serves a different function. The forget gate is responsible for determining whether the information from the previous timestamp should be retained or disregarded. The input gate is responsible for acquiring fresh information from the input [131]. The output gate which sends the new information from current to the next timestamp. This is accomplished via a sigmoid function, which returns a number between zero that is (''totally forget'') and one which is (''completely keep'') when executed. Every time an LSTM network is activated, it creates two states. Those are, a cell state that is passed to the next time-step, as well as time-step's output vector is hidden state. A simple architecture of the LSTM is presented in figure 14.

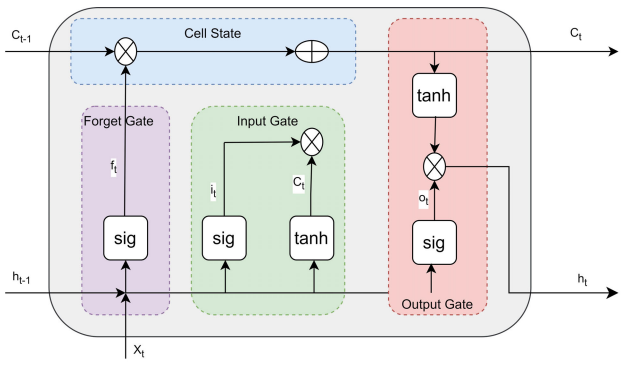

FIGURE 14. Structure of the LSTM.

Since their introduction, several DL based spam detection algorithms have been proposed. Yang and his team outlined an email classification system called Multi-Modal Architecture with Model Fusion (MMA-MF). The primary focus of this model is to identify spam by processing the email's text and images independently using an LSTM model and a CNN model, respectively. An LSTM model is utilized to determine the likelihood that an email is spam based on its textual content. Meanwhile, a CNN model is used to determine the spam likelihood based on any attached images [77]. In another study, a combined model using an LSTM, LR, NB, RF, k-NN, SVM and DT was tested on the UCI SMS spam collection dataset with various embedding techniques (count vectorizer, TF-IDF vectorizer and hashing vectorizer). The highest accuracy of 98.5% was achieved by the LSTM method in this combined architecture [78]. Moreover, a Semantic LSTM (SLSTM) was proposed for spam SMS detection and classification using the SMS Spam Collection dataset and Twitter dataset. The SLSTM incorporates a semantic layer into an LSTM network using Word2Vec word embeddings. Experiments showed the proposed SLSTM technique achieved accuracy results of 99.01% on the SMS Spam Collection dataset and 95.09% on the Twitter dataset [132]. Furthermore, a lightweight GRU (LG-GRU) was employed instead of the LSTM layer for spam classification on the SMS Spam Collection dataset. To improve the semantic understanding of the SMS text inputs, external information from WordNet was incorporated. Compared to LSTM models, the proposed LG GRU model drastically reduced training time and the number of parameters, while maintaining 99.04% accuracy for spam categorization [79]. Additionally, RNNs are one type of NN that can remember past data but suffer from vanishing and exploding gradient issues. To overcome this drawback, the proposed system leverages the Spambase and Ling Spam datasets to classify spam and ham emails using an LSTM architecture. LSTM keeps track of prior email information and learns to select relevant features while ignoring irrelevant ones for identifying spam. Experiments showed the LSTM method achieves 97.4% accuracy, outperforming other DL methods on these datasets [80]. Moreover, spam emails are used for propaganda, advertising, and phishing, which can financially and morally harm internet users as well as disrupt internet traffic. To address this issue, detected spam emails in a Turkish dataset with 100% accuracy using the Keras library and LSTM method. The results demonstrated that an LSTM based method was highly effective for spam detection in Turkish emails [133]. Furthermore, spam emails cause issues like network disruption and cybercrime. A sentiment analysis-friendly spam mail detection method was proposed using Word Embedding techniques including Bag of Words, Hashing, and an LSTM method. Experiments on a dataset of 5,572 messages showed the proposed technique achieved 93-98% in precision, recall, F1-score, and accuracy [134]. The algorithm for email spam detection using an LSTM is presented in algorithm 8.

I. GATED RECURRENT UNIT

GRU is an RNN version that employs gating methods to solve vanishing gradient problem through controlling information flow between cells in the neural network. Kyunghyun Cho introduced the GRU network in 2014, This RNN is almost like LSTM neural network [135]. The structure of the GRU allows it to effectively capture dependencies from large

Algorithm 8 LSTM Algorithm for Email Spam Detection

1: Input Email Spam dataset
2: Convert the text data into numerical vectors using word embeddings
3: Split the data into training and testing
4: Define LSTM architecture
5: Set the LSTM units and hidden layers
6: Add an embedding layer to convert numerical vectors into word embedding
7: Add dropout
8: Add dense output layer using sigmoid
9: Compile with binary cross-entropy
10: Train the method with specified epochs
11: Evaluate the method
12: Predict the email message (spam or ham)

sequences of data in a flexible manner, while retaining knowledge from prior sections of the sequence. The GRU model consists of two gating mechanisms: the update gate and the reset gate [136]. This neural network utilises only one hidden state to concurrently retain both long-term and short-term memory. The reset gate is formulated and calculated by incorporating the hidden state from the previous time step and the input data from the current time step. The gate controls the integration of new input with existing memory [137]. The update gate is used for how much of the previous state is kept. This is extremely useful since the method may choose to duplicate all previous data and remove the possibility of vanishing gradients. This is accomplished via a sigmoid function, which returns a number between 0 and 1. For this simple architecture, the network is able to train rapidly [138]. A simple architecture of the GRU is presented in figure 15.

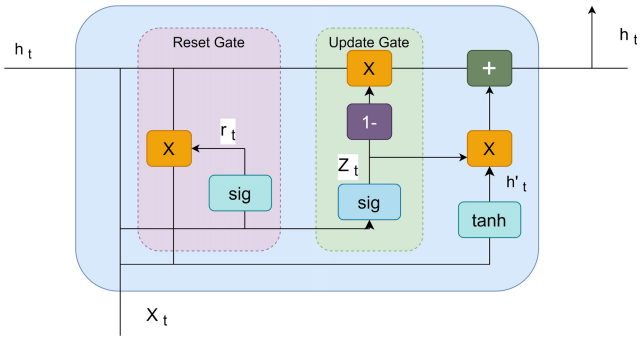

FIGURE 15. Structure of the GRU.

Email spam detection poses a sequence modeling problem well-suited for GRU. A GRU-based architecture for spam detection would process the email text sequentially, encoding each word into a hidden state vector. The gating units in the GRU regulate the flow of information, learning to identify key words and phrases that serve as indicators of spam or legitimate emails [139]. Additionally, as the GRU progresses through the email text, its hidden state captures relevant context and sequentially whether the message is likely to be spam or not. The ability of GRUs to selectively propagate relevant information while processing variable length sequences makes them a promising approach for modeling email text for spam detection [70]. Moreover, a new DL approach uses CNN and RNN to analyze email communication by classifying message components into zones. The method leverages GRU-CRF to segment emails into zones like header, quotation, greeting, and body. Experiments show the technique achieves 98 accuracy on zone prediction, outperforming traditional methods, with improved adaptability and resilience [140]. Furthermore, a lightweight GRU (LG-GRU) was employed instead of an LSTM layer for spam classification on the SMS Spam Collection dataset. To improve the semantic understanding of the SMS text inputs, external information from WordNet was incorporated. Compared to LSTM models, the proposed LG-GRU model drastically reduced training time and the number of parameters, while maintaining 99.04% accuracy for spam categorization [79]. The algorithm for email spam detection using GRU is presented in algorithm 9.

Algorithm 9 GRU Algorithm for Email Spam Detection

1: Input Email Spam dataset
2: Convert the text data into numerical vectors using word embeddings
3: Split the data into training and testing
4: Define GRU architecture
5: Set the GRU units and hidden layers
6: Add an embedding layer to convert numerical vectors into word embedding
7: Add dropout
8: Add dense output layer using sigmoid
9: Compile with binary cross-entropy
10: Train the method with specified epochs
11: Evaluate the method
12: Predict the email message (spam or ham)

J. BIDIRECTIONAL LONG SHORT-TERM MEMORY

Bi-LSTM builds on the standard LSTM architecture to method sequential data more effectively. In contrast to traditional LSTMs that process inputs in only the forward direction, Bi-LSTMs also process the sequence in reverse [141]. This bidirectional approach provides complete past and future context to the method. The Bi-LSTM is composed of two LSTM layers. One layer processes the input sequence in a forward direction, starting from the beginning and ending at the end. The other layer processes the input sequence in a reverse direction, starting from the end and ending at the beginning [114]. The outputs from both directions are concatenated at each time step to generate the final output. This allows the method to preserve contextual information from the entire sequence

when making predictions [142]. A simple architecture of the Bi-LSTM is presented in figure 16.

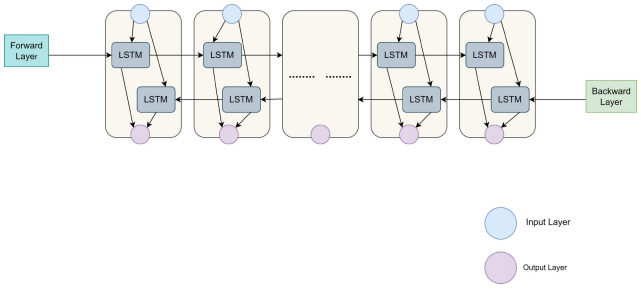

FIGURE 16. Structure of the Bi-LSTM.

The task of email spam detection involves the construction of models that capture the contextual information of words inside an email, enabling the determination of whether the email's content may be classified as spam or not. The Bi-LSTM model is very suitable for this particular task because of its ability to effectively capture both semantic and syntactic links between words. This is achieved by processing the email content in both forward and backward orientations [143]. Additionally, a new DL model for email spam detection using sentiment analysis of email text, combining WordEmbeddings, CNN, and Bi-LSTM networks to analyze textual and sequential properties. Evaluated on two spam datasets, the method achieves improved accuracy of 98-99% and outperforms popular classifiers and state-of-the-art methods, proving its superiority for spam detection [144]. Moreover, spam emails are becoming more common and troublesome as email usage grows, so there is a need for effective methods to detect spam. A recent study compared different ML and DL models, such as RF, NB, ANN, SVM, LSTM, and Bi-LSTM, for the task of identifying spam emails. The study found that Bi-LSTM had the best accuracy of 98.57% for spam prediction [145]. Furthermore, spam text messages steal information from users and hurt them, but the methods available for finding them aren't good enough. The vectorization-based feature engineering and Bi-LSTM networks can be used together to make an effective predictor that can find spam SMS. Experiments showed that the method is more accurate than other methods in terms of precision, recall, and F1 measures [146]. The algorithm for email spam detection using Bi-LSTM is presented in algorithm 10.

The LSTM model has proven to be the most effective for email spam detection due to its specialized architecture designed for sequential data. Emails are inherently sequential, consisting of words and sentences in a specific order, which aligns perfectly with LSTM's strengths. The model's memory cell excels at capturing long-term dependencies and contextual information, allowing it to effectively learn patterns and relationships between words or tokens in email sequences. This ability to retain and process contextual information over many timesteps is crucial for spam detection, as important clues may be spread throughout the email body.

Algorithm 10 Bi-LSTM Algorithm for Email Spam Detection

1: Input Email Spam dataset
2: Convert the text data into numerical vectors using word embeddings
3: Split the data into training and testing
4: Define Bi-LSTM architecture
5: Set the Bi-LSTM units and hidden layers
6: Add an embedding layer to convert numerical vectors into word embedding
7: Add dropout
8: Add dense output layer using sigmoid
9: Compile with binary cross-entropy
10: Train the method with specified epochs
11: Evaluate the method
12: Predict the email message (spam or ham)

LSTM's adaptability to various writing styles and content types further enhances its effectiveness across different datasets and evolving spam techniques.

To further improve LSTM's accuracy in email spam detection, several modifications can be considered. Incorporating attention mechanisms could help the model focus on the most relevant parts of an email. Ensemble methods, combining LSTM with other models, could leverage the strengths of different approaches. Transfer learning, by pre-training the LSTM model on a large corpus of email data, could enhance performance, especially when dealing with limited labeled data. Additional strategies such as feature engineering, regularization techniques, hierarchical LSTM structures, and character-level input processing could also contribute to improved accuracy.Furthermore, numerous evaluation metrics have been employed to measure the effectiveness of these LSTM model. Here are some frequently used metrics in the papers we have reviewed:

Accuracy: Accuracy is one factor to consider when rating categorization models. Accuracy is the proportion of forecasts that method predicted successfully. For binary classification, accuracy can also be assessed in terms of positives and negatives, as shown below:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Precision: Precision can also be used to judge how well an identifying system works. It is found by adding up the number of true positives to the number of fake positives for each class. It shows really good cases out of all the optimistic forecasts.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

Recall: Recall is a quantitative measure that indicates the proportion of instances correctly identified by the method among all the possible positive labels. The term refers to the ratio of true positive cases to the sum of true positive and false

negative cases.

$$\text{Recall} = TPR = \frac{TP}{TP + FN} \tag{3}$$

F1-score: The accuracy metric quantifies the frequency at which a model accurately predicted the entirety of the dataset.

$$F1\text{-score} = 2 * \left(\frac{\text{Precision*Recall}}{\text{Precision + Recall}} \right) \tag{4}$$

IV. DATASETS COLLECTION AND PRE-PROCESSING

A. DATASETS

The collection of data samples contained within a corpus plays a pivotal role in evaluating the efficacy of any spam detection technique. While there exists several conventional datasets that are commonly leveraged to assess text classification, only recently have researchers publishing new spam detection methodologies made an effort to provide public access to the same corpora of emails applied to assess the effectiveness of their proposed methods. A comprehensive listing of publicly released spam email datasets referenced across the datasets characterize covered in this paper are summarized in Table 5. Each corpus contains intrinsically unique traits and labeling that ultimately dictate the generalizability and alignment of experimental outcomes for every published approach utilizing that data source. Key dimensions that characterize an evaluation dataset's nature include the size of emails, proportional class balance between spam and ham samples.

The vast majority of features leveraged to distinguish spam from legitimate emails manifest in textual content. Applying appropriate pre-processing to standardize, clean, and filter this text data represents a foundational data wrangling step prior to method development. The following sub-section provides the details of pre-processing techniques.

B. PRE-PROCESSING TECHNIQUES

Before data can be analyzed, it must be prepared through a process called preprocessing. Raw datasets often contain inconsistencies like missing values, duplicate entries, and text in incompatible formats that methods cannot interpret. Preprocessing transforms messy raw data into a clean form that analytical methods can work with effectively. This crucial step improves the accuracy of later analysis. Common preprocessing tasks include handling incomplete data, standardizing text into numerical forms, extracting informative features, and removing noise. Careful preprocessing allows methods to discover more robust patterns and make better predictions. Mostly used preprocessing techniques for email spam detection is given below:

1) HANDLING MISSING VALUES

The management of missing values in datasets is a key component in preventing bias and ensuring that methods continue to produce accurate results. There are a number of approaches that can be utilized, including the elimination of missing numbers or the substitution of such values with the mean, the median, or specified values.

C. TEXT PREPROCESSING

Text preprocessing transforms raw text data into a cleaner form before analysis. Removing extraneous elements allows more accurate feature extraction and developing further downstream. Preprocessing is thus an essential first step when working with text data. Common text cleaning tasks include stripping punctuation, deleting HTTP links, eliminating special characters, getting rid of stop words, lowercasing all text, correcting spellings, and more. Numerous text-preprocessing techniques exist for the purpose of eliminating unnecessary information from incoming text input, as shown in Figure 17.

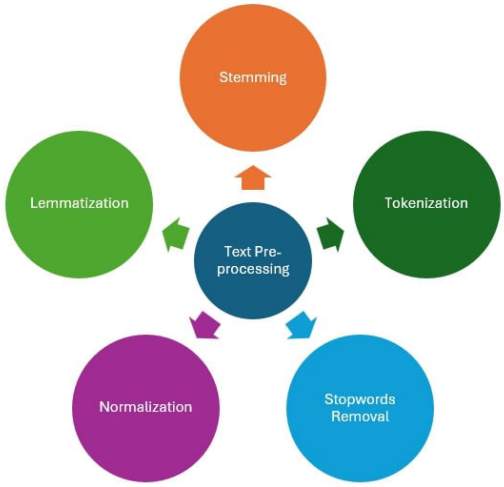

FIGURE 17. Various text preprocessing techniques.

1) STEMMING

Stemming seeks to simplify text analysis by stripping words down to their base form. Tools match terms like "drunk", "drink", and "drank" to their core stem - "drink". This normalization groups together different inflections, allowing more generalized patterns to emerge. Stemmers remove suffixes systematically using rule-based algorithms like the popular Porter stemmer in Python's NLTK library. However, overly zealous stemming risks both under stemming and over stemming textual data. Under stemming fails to fully reduce related terms down to one stem.

2) TOKENIZATION

Tokenization splits text into discrete units for analysis. First, extraneous characters like HTML and punctuation are filtered out. Then words and numbers are extracted into individual tokens by splitting on whitespace and symbols. These atomic elements can be manipulated, counted, classified and more. Tokenization forms the basis for quantitative text analysis. This preprocessing step makes linguistic features accessible using Python's Regex library and Natural Language Processing toolkits. Proper tokenization increases performance

TABLE 5. Publicly available email spam datasets.

References	Datasets Name	Spam Rate	Message Number		Creation Year
			Spam	Ham	
[7]	Spamemail	32%	1378	2949	2010
[147]	Hunter	53%	928	810	2008
[148]	Dredze image spam Dataset	62%	3297	2021	2007
[149]	Princeton spam image Benchmark	100%	1071	0	2007
[150]	Trec 2007	67%	50,199	25,220	2007
[151]	Trec 2006	66%	24,912	12,910	2006
[152]	Enron-spam	55%	20170	16545	2006
[153]	Phishing corpus	100%	415	0	2005
[153]	Biggio	100%	8549	0	2005
[154]	Trec 2005	57%	52,790	39,399	2005
[155]	Gen spam	78%	31,196	9212	2005
[156]	Zh1	74%	1205	428	2004
[156]	PUA	50%	571	571	2003
[156]	PU3	44%	1826	2313	2003
[156]	PU2	20%	142	579	2003
[157]	Spamassassin	31%	1897	4150	2002
[158]	PU1	44%	481	618	2000
[159]	Lingspam	17%	481	2412	2000
[159]	Spambase	39%	1813	2788	1999
[160]	Spam archive	100%	15090	0	1998

on tasks ranging from sentiment classification to document summarization. Table 6 shows a sample sentence and its associated tokens.

3) STOPWORDS REMOVAL

Stop words are common filler words that carry little meaning, such as ''a'', ''an'', ''so'', ''and'', and ''the''. Though frequently occurring, these terms contribute more noise than signal during text analysis. Filtering out stop words shrinks datasets down to more meaningful vocabulary. Most text analysis toolkits provide standard stop word lists and functions like Python's NLTK library to effortlessly strip this cover. Table 7 presents the descriptions and web URLs of several libraries and packages that are accessible for the purpose of preprocessing text data.

4) NORMALIZATION

Normalization transforms text into a standard format to enhance analysis. This preprocessing step structures messy linguistic data by correcting variant spellings, coercing case and tense, resolving contractions, converting numbers to numerals, transliterating terms, aligning related words to a root form via stemming and lemmatization, and more.

5) LEMMATIZATION

Lemmatization maps words to their root form using lexical analysis. It relies on dictionaries and knowledge of morphology to connect related terms to the same base lemma. For example, the words ''plays'', ''playing'', and ''played'' all share the lemma ''play''. Lemmatizers can thus group together different inflections and variants by canonicalizing them to their common origin. Tools like NLTK's WordNet Lemmatizer leverage semantic databases to correctly resolve words to their underlying lemma based on context. Properly deploying lemmatization avoids incorrectly collapsing unrelated words while clustering together meaningful word associations, boosting performance on semantics-sensitive tasks.

D. FEATURE EXTRACTION TECHNIQUES

Feature extraction converts unstructured text into quantitative data amenable to modeling, by transforming documents into numerical vectors. Common methods calculate Term Frequency-Inverse Document Frequency (TF IDF) weights, Bag of words (BoW), count N-gram patterns, encode syntactic Parsing Trees, apply Topic Modeling algorithms like Latent Dirichlet Allocation, or ingest word vectors (Word2Vec). Robust text analytics combines multiple feature extraction methods to fully capture linguistic complexity within interpretable data structures.

Spam is a major issue in current email communication, stemming from motives like advertising and fraud. To effectively detect spam, appropriate preprocessing techniques are needed, such as removing noise, taking out common stop words, stemming, lemmatization, and adjusting term frequencies. Mallampat et al. proposed a multi-modal system (MMA FM) that uses a combined method (IMTF-IDF+Skip-thoughts) and a CNN to extract features. This achieves superior 99.16% accuracy in identifying spam compared

TABLE 6. A representation of a sentence and the tokens it automatically generates.

Sentence	Tokens
I am playing cricket in the school field	"I", "am", "playing", "cricket", "in", "the", "school", "field"

TABLE 7. Pre-processing tools for text.

Package	Description	Link
NLTK	The Natural Language Toolkit, abbreviated as NLTK, is a collection of Python-based tools and programs designed for the purpose of executing natural language processing (NLP) tasks.	https://www.nltk.org/
TextBlob	TextBlob is a Python library used for processing text. The API offers a direct and uncomplicated interface for common NLP operations, such as assigning parts of speech to words and analyzing sentiment.	https://textblob.readthedocs.io/en/dev/
RapidMiner	The process of accessing and analyzing many forms of data, including both organized and unstructured data, is made easier.	https://rapidminer.com/products/studio/feature-list/
Spacy	Spacy is a Python module for NLP that includes several pre-installed functionalities.	https://spacy.io/
Memory-Based Shallow Parser	Able to use Python to parse a string of letters or words and determine the sentence's grammatical structure	https://pypi.org/project/MBSP-for-Python

to using Naive Bayes, when tested on the Enron, Dredze, and TREC 2007 datasets [161]. Saini et al. introduced a new method for predicting email spam that uses random forest for feature extraction. The features extracted by the random forest are then fed into a logistic regression method which predicts whether an email is spam [162]. Cheng et al. presented a new attack method that strategically modifies text data using insights from adversarial examples. It intentionally alters features that represent an email. They explored different feature extraction techniques using various NLP methods. Their study designs effective mechanisms to translate adversarial perturbations back into magic words"in the text. This causes intentional misclassifications across multiple datasets and ML methods under white-box, gray-box and black-box attack scenarios [163]. Hassan et al. tested different feature extraction techniques along with two supervised ML classifiers on two public spam email datasets. They emphasized the importance of finding the optimal pairing of feature extraction and classification method. They also highlighted the benefits of testing on different datasets. SVM and NB showed impressive accuracy with TF-IDF, reaching over 99% and around 98% respectively [164]. Table 8 presents the previous research on spam detection using feature extraction techniques.

1) BAG OF WORDS (BOW)

BoW representation is a simple yet powerful approach for extracting numeric text features. This method counts the occurrences of words within a document while disregarding grammar and word order. Documents become vectors denoting the frequency of terms like ''cat'', ''tree'', and ''slept''. Bags-of-words thus efficiently quantifies unstructured text as matrices tallying vocabulary. Many extensions enrich this basic technique like n-grams counting multi-word expressions and skip-grams sampling non-contiguous patterns. For instance, Barushka et al. detected deceptive hotel reviews on TripAdvisor by representing documents as n-gram frequencies and skip-gram embeddings to train machine learning classifiers. Bag-of-words style features unlock effective text analysis despite ignoring complex linguistic structure [170]. The flexibility of multiple vocabulary quantification strategies enables customized feature engineering for tasks ranging from spam detection to sentiment analysis across domains.

2) ONE HOT ENCODING

One-hot encoding transforms text into numeric features by assigning each unique word or token its own binary vector. Documents represent bags of these orthogonal hot vectors - sparse yet unambiguous codes with a single ''1'' marking the presence of each distinctive term. One hot encoding matrices efficiently quantify textual data, with vector lengths equal to vocabulary size rather than the longer original raw text. By indexing words into binary indicator columns, this method facilitates quantitative analysis while retaining the ability to map patterns back to original tokens. One hot encoding forms the input for many machine learning algorithms, often outperforming methods lacking explicit word-level encoding. The simplicity of tallying vocabulary into orthogonal dimensions makes one hot representation a widely useful feature extraction technique for textual data.

3) WORD EMBEDDING

One-hot encoding scales poorly to large vocabularies due to its explosion of sparse binary features. Embedding methods address this weakness through distributed representation. Word embeddings map language into compact dense vectors capturing similarities between related terms. For instance, vectors for cat and kitten cluster together, unlike the orthogonal one-hot encoding. This efficiency facilitates DL on extensive corpora. Embeddings also encode meaning - algebraic operations reveal relationships like king is to queen as man is to woman. Created using neural networks, embeddings represent both syntax and semantics within a low-dimensional subspace. Methods learn contextual associations, quantifying elusive concepts like gender or formality. Versatile representations power cutting-edge applications from chatbots to search. Embed-

dings distill enormous dictionaries into meaningful, manipulable codes advancing the frontiers of text mining.

4) WORD2VEC

This method turns words into vectors and works like a two-layer network to handle text that is made up of words. There is a matched vector in the space for every word in the corpus. Either a continuous skipgram or a continuous bag of words design (CBOW) is used by Word2vec. In the case of the continuous skipgram, the present word is used to guess the words that come after it. In the CBOW method, on the other hand, the surrounding or neighboring words are used to guess a middle word. With a small amount of training data, the skip-gram method can correctly represent even rare words or phrases. However, the CBOW method is many times faster to train and is a little more accurate for common keywords. The word2vec method is better because it lets you learn high-quality word embedding in less time and space. From a much larger body of writing, it is possible to learn bigger embeddings (with more dimensions).

5) N-GRAMS

A lot of Natural Language Processing (NLP) tasks use N-grams, which are long strings of words or tokens in a text. Based on the number of "n," they are divided into different groups, such as Unigram, Bigram, and Trigram. Kanaris et al. used a set of 2,893 emails to pull out n-gram traits from text. In their study, they looked at success factors like spam recall and precision. Combining SVM with n-grams, they were able to make a spam filtering method that had an accuracy score of more than 0.90 for finding spam [171]. Table 9 below shows several examples of N-grams.

6) TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF)

The BoW approach faces challenges with high-frequency terms dominating the data while lower-scoring domain-specific words may be eliminated or ignored. An improvement on bag-of-words is the TF-IDF technique. TF-IDF multiplies the number of times a term appears in a document TF by the inverse of how often that term shows up across all documents (inverse document frequency or IDF). These scores highlight unique and information-rich terms within a document. As demonstrated by Equations 1 and 2, TF represents the ratio of the term's count in the document to the overall count of all terms. On the other hand, IDF is the logarithm of the total number of documents divided by the number of documents that contain the term. The resulting TF-IDF scores better represent a term's significance.

$$\mathrm{Tf}(w) = \frac{\text{Number of times word } w \text{ appears in a document}}{\text{Total number of terms in the document}} \tag{5}$$

$$\mathrm{Idf}(w) = \log_e \frac{\text{Total number of documents}}{\text{Number of documents with term } w} \tag{6}$$

7) GLOVE WORD EMBEDDING

This is an unsupervised method that generates a vector to represent words or text. It aims to capture semantic and contextual meaning of words. It is a count-based method that utilizes co-occurrence statistics of words in a corpus. Specifically, it trains on the non-zero entries in a word-context co-occurrence matrix. The key intuition behind Glove Word Embedding is that ratios of word-word co-occurrence probabilities can encode meaning. Equation 3 demonstrates the computation of the co-occurrence probability for the texts in each word embedding.

$$V(tx, ty, tz) = \left(F_{xy}/F_{yz}\right) \tag{7}$$

where,

- The co-occurrence possibility for the texts tx and ty is F_{xy}.
- The co-occurrence possibility for the texts ty and tz is F_{yz}.
- The regular texts or words that appear in a document are tx and ty and the investigated text is tz.
- When the above-mentioned ratio is 1, the investigated text is related to tx rather than ty.

V. IMPLICATIONS

The review covered a comprehensive analysis and integration of the present condition of email spam detection. A broad range of ML and DL approaches for email spam detection is covered, along with an analysis of how these approaches could be improved for greater efficiency. The review explored the intricate difficulties encountered in identifying and screening spam emails while recognizing the constraints of conventional techniques such as blocklists, real-time blackhole listing, and content-based approaches. The review analyzed and addressed current research deficiencies, shedding light on areas that require additional exploration. This will emphasize the continuous necessity for innovation and enhancement in spam detection techniques. In addition, the study suggested potential areas for future research, highlighting possible paths for further advancement and directing researchers towards addressing the observed deficiencies.

The review emphasized the importance of effective spam detection in order to safeguard users from the detrimental effects of spam, including time wastage, resource depletion, and potential data theft, given the widespread use of emails across many industries. The objective of the study is to offer a methodical and empirically supported comprehension of current research, assessing the efficacy of various ML and DL techniques. Through the synthesis and examination of data from many studies, it aims to provide an impartial assessment of the advantages and disadvantages of current methodologies. The thorough assessment of methods for identifying email spam has substantial ramifications for the domains of digital communication and cybersecurity. The study examined the application of various ML and DL techniques, with a focus on shifting from traditional

TABLE 8. Feature extraction techniques for email spam.

Reference	Datasets	Merit	Methods	Limitation	Results
[165]	Honeypot, SPD manually and automatically annotated spam dataset	Real-time detection of spam is feasible, and the suggested feature set enhances the accuracy of the system.	SVM, RF, MLP and GB	Require a solution to address the occurrence of extended tweets related to spamming behavior.	Accuracy: 97.71% Precision: 99% Recall: 97% F1-score: 98%
[166]	13,000 comments from YouTube channels	The utilization of N-grams in ML methods has significantly enhanced the accuracy of classification.	RF, SVM, NB with N-grams based features	Utilizing advanced word representation techniques such as Word2Vec is necessary in order to enhance the efficiency of the system.	F1-score: 97%
[167]	More than 10,000 Arabic tweets collected with Twitter API	The time required to classify the tweets is significantly lower compared to the most advanced algorithms available.	LSTM with word embedding feature representation	The accuracy of system classification is depending upon the length of the tweet.	Accuracy: 97% Precision: 98% Recall: 95% F1-score: 97%
[168]	97,839 Restaurant and 31,317 Hotel review dataset	Multimodal neural network method could catch sophisticated spammer actions.	CNN and Bi-LSTM	It is necessary to examine the utilization of more efficient characteristics in order to enhance the performance.	Precision: 82% Recall: 80% F1-score: 81%
[169]	10 day real-life Twitter dataset of 1,376,206 spam and 6,73,836 non-spam tweets	Changes in spamming operations are detected immediately.	RF, MLP and NB	The method must possess the ability to easily adjust to new attributes.	Accuracy: 99.35% Precision: 95.84% Recall: 91.03% F1-score: 93.37%

methodologies to more sophisticated ones. This change has the capacity to improve the precision of detection and the efficiency of computing. This technological advancement may lead to enhanced email systems that offer more robust defenses against harmful material and reduce the wasteful consumption of resources. The review comprised a comprehensive analysis and integration of the present condition of email spam detection.

VI. CHALLENGES OF EMAIL SPAM DETECTION

Spam detection systems have difficulty figuring out how to properly evaluate features across textual, temporal, semantic, and statistical dimensions because the amount of different and complicated data on the Internet is growing all the time. Additionally, most methods are trained on balanced datasets which rarely match real-world conditions. Self-learning methods that can adapt without manual supervision remain an open area. Spam detection methods also face various adversarial attacks - poisoning attacks that pollute training data, evasive attacks that manipulate test samples to bypass filters, and privacy attacks attempting to steal sensitive training data. Deep fakes leveraging AI generation and modification techniques around images, video and text for disseminating misinformation further threaten credibility.

TABLE 9. Illustration of an n-grams.

SL.No	Type of N-Gram	Example
1	Uni-gram	"I", "love", "to", "eat", "fried", "rice"
2	Bi-gram	I love, to eat, fried rice
3	Tri-gram	I love to, eat fried rice

Imbalanced datasets with far more legitimate emails than spam continue biasing method performance towards false positives. Research on intelligent oversampling methods aims to improve minority class representation during training. The dynamic evolution of spam tactics also reduces generalization capabilities against new previously unseen attacks. Ensuring method robustness through adversarial training is an active

research direction. Potential adversarial samples crafted specifically to fool deep nets pose reliability hurdles. Detecting adversarial patterns and training on adversarial datasets helps improve resilience.

The black-box nature of deep networks also hampers method interpretability and user trust. Advancing explainable AI to increase transparency in method behaviors and decisions thus remains important. The computational intensity for large-scale DL limits accessibility to organizations with fewer resources, though optimizations around method efficiency and hardware acceleration are progressively lowering barriers.

Generalizability across different email systems, user groups, and usage patterns is needed for wide real-world deployment. Multi-model learning and personalization are promising techniques under investigation. Adoption is made harder by problems with privacy, usability, and integrating content analysis across a wide range of old systems and infrastructure. Limited availability of labeled data for adequately training deep nets continues to be an industry wide bottleneck, although data augmentation, transfer learning and semi-supervised techniques help multiply value from limited labels. Finally, meeting real-time latency demands at scale for live traffic with deep methods has throughput and optimization implications. Quantization, network pruning and efficient method distillation actively aim to improve inferencing speed.

VII. RESEARCH GAPS AND OPEN RESEARCH PROBLEMS

This section examines the areas where research is lacking and the problems that remain in the field of spam identification. Current detection approaches rely heavily on manually engineered datasets which rarely match the nuanced complexities of real-world spam. Future work should select developing robust methods using authentic spam samples only. Though ML, fuzzy logic and DL methods are individually leveraged today, hybrid systems that synergistically combine multiple techniques could potentially improve accuracy and efficiency further. Enhanced feature engineering leveraging deep neural networks' self-learned representations via representation learning presents promising opportunities to automatically capture differentiating attributes. Clustering algorithms that enable dynamic spam database updates based on continuous user feedback requires exploration for tighter spam relevancy. In addition to DL based blockchain methods and concepts can potentially be employed for email spam detection in the future. Advancing the art of manual spam dataset annotation by collaborating with linguistics and psychology experts can potentially better encapsulate semantic and cognitive subtleties within messages for training more discerning models. Hardware optimizations leveraging graphics cards and field-programmable gate arrays provide additional vectors to improve real-time throughput and latency when classifying high-velocity email streams. Centrally, the availability of multifaceted, standardized labeled corpora spanning diverse, real-world spam types remains lacking, constraining more robust solutions. Furthermore, it is essential for future research to focus on providing researchers with standardized labelled datasets to train classifiers. Additionally, enhancing the accuracy and reliability of spam detection methods can be achieved by incorporating other features into the dataset, such as the spammer's IP address and location. The following are further fields of future study and open problems that need to be solved in the field of spam detection:

Current spam detection approaches rely heavily on limited features from email headers, subject lines, and message bodies. To improve accuracy, more comprehensive and automated feature engineering is needed, moving beyond manual selection. While most evaluations focus on statistical performance metrics like precision and recall, incorporating time complexity analysis would provide crucial insight into real-world viability. Exploring advanced feature extraction methods using DL on various email components, beyond just message bodies, can reveal more nuanced signals for detection. Several system design aspects warrant focus to enhance practical applicability. These include improving fault tolerance for reliability, ensuring quick response times under heavy loads, and implementing self-learning capabilities without manual supervision for robust adaptability to evolving spam tactics. Dynamic updating of feature representations using deep neural networks as new spam data emerges can bolster detection relevance over time. Ensuring strong security mechanisms against exploratory attacks or poisoning of the pipeline data or model itself is imperative for trustworthy operation. Reducing false positive rates continues to pose challenges to usability. Expanding beyond textual spam to effectively flag image-based messages and addressing real-time threats rather than relying on batch processing, given the low latency constraints of email systems, will significantly expand practical applicability. Several promising research directions emerge. The lack of labeled multilingual corpora presents an opportunity for developing more globally effective solutions. Semi-supervised learning methods could help leverage vast amounts of unlabeled data. Identifying coordinated spammer networks and behaviors could lead to more proactive defense strategies. Rather than manual labeling or curation that can introduce unconscious bias, discovering ground truth spam characteristics automatically through federated learning over decentralized data holds potential for more robust and unbiased models. Exploring the potential of large language models in transforming spam detection is justified due to their ability to catch intricate patterns and contextual nuances that conventional methods may overlook. Studying the potential of fine-tuning pre-trained models such as BERT or GPT for spam classification tasks could lead to the development of more precise and flexible spam detection systems. Moreover, the utilization of these expansive models could potentially tackle existing obstacles in spam detection, including managing evolving spam strategies and minimizing false positives, hence facilitating the development of more resilient and effective spam detection solutions.

VIII. CONCLUSION

The purpose of this literature review is to provide a summary of the most recent research on the application of ML and DL for the detection of spam in email. It provides illuminates on a number of shortcomings as well as potential enhancements that could be made to enhance the efficiency of detection against constantly developing spammer strategies. The implementation of detection systems in close proximity to primary servers, expanding beyond linguistic analyses, and broadening the scope of content evaluation are all examples of prospective advancements. Prioritization is crucial for effectively addressing modern attack types, managing concept drift, enhancing model generalizability, and aligning training with test performance. The report examines the evolution of machine learning and deep learning applications in distinguishing spam from legitimate communications, particularly in the context of spammers circumventing existing filters. By comparing various methodologies and highlighting unresolved research challenges, the report illuminates persistent difficulties in this field. While current state-of-the-art approaches have limitations, focused efforts on recommended improvements can significantly enhance both accuracy and efficiency. Future research can be directed towards identified shortcomings to develop more robust anti-spam systems. The synthesized insights enable researchers to refine spam protection strategies through meticulous enhancements that proactively address both current and emerging threats. Key areas for improvement include adapting to evolving attack patterns, mitigating concept drift in spam detection models, improving model generalizability across diverse communication contexts, and reducing discrepancies between training and real-world performance. By concentrating on these aspects, researchers can create more effective and adaptable anti-spam solutions that stay ahead of sophisticated spam tactics.

REFERENCES

[1] K. Deshpande, J. Girkar, and R. Mangrulkar, "Security enhancement and analysis of images using a novel sudoku-based encryption algorithm," *J. Inf. Telecommun.*, vol. 7, no. 3, pp. 270–303, Jul. 2023.

[2] D. Goel and A. K. Jain, "Mobile phishing attacks and defence mechanisms: State of art and open research challenges," *Comput. Secur.*, vol. 73, pp. 519–544, Mar. 2018.

[3] J. Doshi, K. Parmar, R. Sanghavi, and N. Shekokar, "A comprehensive dual-layer architecture for phishing and spam email detection," *Comput. Secur.*, vol. 133, Oct. 2023, Art. no. 103378.

[4] F. Salahdine and N. Kaabouch, "Social engineering attacks: A survey," *Future Internet*, vol. 11, no. 4, p. 89, Apr. 2019.

[5] M. Alawida, A. E. Omolara, O. I. Abiodun, and M. Al-Rajab, "A deeper look into cybersecurity issues in the wake of COVID-19: A survey," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 10, pp. 8176–8206, Nov. 2022.

[6] B. Parmar, "Protecting against spear-phishing," *Comput. Fraud Secur.*, vol. 2012, no. 1, pp. 8–11, Jan. 2012.

[7] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, "Machine learning for email spam filtering: Review, approaches and open research problems," *Heliyon*, vol. 5, no. 6, Jun. 2019, Art. no. e01802.

[8] (2023). *Statista*. [Online]. Available: https://www.statista.com/statistics/456500/daily-number-of-e-mails-worldwide/

[9] O. Fonseca, E. Fazzion, I. Cunha, P. H. B. Las-Casas, D. Guedes, W. Meira, C. Hoepers, K. Steding-Jessen, and M. H. P. Chaves, "Measuring, characterizing, and avoiding spam traffic costs," *IEEE Internet Comput.*, vol. 20, no. 4, pp. 16–24, Jul. 2016.

[10] S. Ogwu, P. Sice, S. Keogh, and C. Goodlet, "An exploratory study of the application of mindsight in email communication," *Heliyon*, vol. 6, no. 7, Jul. 2020, Art. no. e04305.

[11] O. A. Okunade. (2017). *Manipulating e-mail Server feedback for Spam Prevention*. [Online]. Available: https://www.azojete.com.ng

[12] (2023). *Firms*. Accessed: Dec. 28, 2023. [Online]. Available: https://99firms.com/blog/spam-statistics/

[13] S. Dhanaraj and V. Karthikeyani, "A study on e-mail image spam filtering techniques," in *Proc. Int. Conf. Pattern Recognit., Informat. Mobile Eng.*, Feb. 2013, pp. 49–55.

[14] A. Bhowmick and S. M. Hazarika, "Machine learning for e-mail spam filtering: Review, techniques and trends," 2016, *arXiv:1606.01042v1*.

[15] C. Laorden, X. Ugarte-Pedrero, I. Santos, B. Sanz, J. Nieves, and P. G. Bringas, "Study on the effectiveness of anomaly detection for spam filtering," *Inf. Sci.*, vol. 277, pp. 421–444, Sep. 2014.

[16] N. Ahmed, R. Amin, H. Aldabbas, D. Koundal, B. Alouffi, and T. Shah, "Machine learning techniques for spam detection in email and IoT platforms: Analysis and research challenges," *Secur. Commun. Netw.*, vol. 2022, pp. 1–19, Feb. 2022.

[17] S. Gibson, B. Issac, L. Zhang, and S. M. Jacob, "Detecting spam email with machine learning optimized with bio-inspired metaheuristic algorithms," *IEEE Access*, vol. 8, pp. 187914–187932, 2020.

[18] T. Gangavarapu, C. D. Jaidhar, and B. Chanduka, "Applicability of machine learning in spam and phishing email filtering: Review and approaches," *Artif. Intell. Rev.*, vol. 53, no. 7, pp. 5019–5081, Oct. 2020.

[19] S. Zavrak and S. Yilmaz, "Email spam detection using hierarchical attention hybrid deep learning method," *Expert Syst. Appl.*, vol. 233, Dec. 2023, Art. no. 120977.

[20] P. K. Roy, J. P. Singh, and S. Banerjee, "Deep learning to filter SMS spam," *Future Gener. Comput. Syst.*, vol. 102, pp. 524–533, Jan. 2020.

[21] S. Magdy, Y. Abouelseoud, and M. Mikhail, "Efficient spam and phishing emails filtering based on deep learning," *Comput. Netw.*, vol. 206, Apr. 2022, Art. no. 108826.

[22] S. Kaddoura, G. Chandrasekaran, D. Elena Popescu, and J. H. Duraisamy, "A systematic literature review on spam content detection and classification," *PeerJ Comput. Sci.*, vol. 8, p. e830, Jan. 2022.

[23] T. Lin, D. E. Capecci, D. M. Ellis, H. A. Rocha, S. Dommaraju, D. S. Oliveira, and N. C. Ebner, "Susceptibility to spear-phishing emails," *ACM Trans. Comput.-Hum. Interact.*, vol. 26, no. 5, pp. 1–28, Oct. 2019.

[24] K. Thakur, M. L. Ali, M. A. Obaidat, and A. Kamruzzaman, "A systematic review on deep-learning-based phishing email detection," *Electronics*, vol. 12, no. 21, p. 4545, Nov. 2023.

[25] R. Li, Z. Zhang, J. Shao, R. Lu, X. Jia, and G. Wei, "The potential harm of email delivery: Investigating the HTTPS configurations of webmail services," *IEEE Trans. Dependable Secur. Comput.*, vol. 21, no. 1, pp. 1–14, Aug. 2023.

[26] A. Abayomi-Alli, O. Abayomi-Alli, S. Misra, and L. Fernandez-Sanz, "Study of the yahoo-yahoo hash-tag tweets using sentiment analysis and opinion mining algorithms," *Information*, vol. 13, no. 3, p. 152, Mar. 2022.

[27] S. A. Ebad, "Lessons learned from offline assessment of security-critical systems: The case of microsoft's active directory," *Int. J. Syst. Assurance Eng. Manage.*, vol. 13, no. 1, pp. 535–545, Feb. 2022.

[28] A. Kumar, "An empirical examination of the effects of design elements of email newsletters on consumers' email responses and their purchase," *J. Retailing Consum. Services*, vol. 58, Jan. 2021, Art. no. 102349.

[29] V. Y. Oviedo and J. E. Fox Tree, "Meeting by text or video-chat: Effects on confidence and performance," *Comput. Hum. Behav. Rep.*, vol. 3, Jan. 2021, Art. no. 100054.

[30] M. K. Islam, M. A. Amin, M. R. Islam, M. N. I. Mahbub, M. I. H. Showrov, and C. Kaushal, "Spam-detection with comparative analysis and spamming words extractions," in *Proc. 9th Int. Conf. Rel., INFOCOM Technol. Optim.*, Sep. 2021, pp. 1–9.

[31] F. Jáñez-Martino, R. Alaiz-Rodríguez, V. González-Castro, E. Fidalgo, and E. Alegre, "A review of spam email detection: Analysis of spammer strategies and the dataset shift problem," *Artif. Intell. Rev.*, vol. 56, no. 2, pp. 1145–1173, Feb. 2023.

[32] N. Pérez-Díaz, D. Ruano-Ordás, F. Fdez-Riverola, and J. R. Méndez, "SDAI: An integral evaluation methodology for content-based spam filtering models," *Expert Syst. Appl.*, vol. 39, no. 16, pp. 12487–12500, Nov. 2012.

[33] N. Saidani, K. Adi, and M. S. Allili, "A semantic-based classification approach for an enhanced spam detection," *Comput. Secur.*, vol. 94, Jul. 2020, Art. no. 101716.

[34] Z. Zhang, E. Damiani, H. A. Hamadi, C. Y. Yeun, and F. Taher, "Explainable artificial intelligence to detect image spam using convolutional neural network," in *Proc. Int. Conf. Cyber Resilience (ICCR)*, Oct. 2022, pp. 1–5.

[35] A. Hosseinalipour and R. Ghanbarzadeh, "A novel approach for spam detection using horse herd optimization algorithm," *Neural Comput. Appl.*, vol. 34, no. 15, pp. 13091–13105, Aug. 2022.

[36] M. Novo-Lourés, D. Ruano-Ordás, R. Pavón, R. Laza, S. Gómez-Meire, and J. R. Méndez, "Enhancing representation in the context of multiple-channel spam filtering," *Inf. Process. Manage.*, vol. 59, no. 2, Mar. 2022, Art. no. 102812.

[37] Z. F. Sokhangoee and A. Rezapour, "A novel approach for spam detection based on association rule mining and genetic algorithm," *Comput. Electr. Eng.*, vol. 97, Jan. 2022, Art. no. 107655.

[38] A. R. Yeruva, D. Kamboj, P. Shankar, U. S. Aswal, A. K. Rao, and C. S. Somu, "E-mail spam detection using machine learning—KNN," in *Proc. 5th Int. Conf. Contemp. Comput. Informat. (IC3I)*, Dec. 2022, pp. 1024–1028.

[39] M. A. Shaaban, Y. F. Hassan, and S. K. Guirguis, "Deep convolutional forest: A dynamic deep ensemble approach for spam detection in text," *Complex Intell. Syst.*, vol. 8, no. 6, pp. 4897–4909, Dec. 2022.

[40] M. F. Faisal, M. N. U. Saqlain, M. A. S. Bhuiyan, M. H. Miraz, and M. J. A. Patwary, "Credit approval system using machine learning: Challenges and future directions," in *Proc. Int. Conf. Comput., Netw., Telecommun. Eng. Sci. Appl. (CoNTESA)*, 2021, pp. 76–82.

[41] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surveys*, vol. 34, no. 1, pp. 1–47, Mar. 2002.

[42] M. RAZA, N. D. Jayasinghe, and M. M. A. Muslam, "A comprehensive review on email spam classification using machine learning algorithms," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2021, pp. 327–332.

[43] N. Govil, K. Agarwal, A. Bansal, and A. Varshney, "A machine learning based spam detection mechanism," in *Proc. 4th Int. Conf. Comput. Methodologies Commun. (ICCMC)*, Mar. 2020, pp. 954–957.

[44] C. Bansal and B. Sidhu, "Machine learning based hybrid approach for email spam detection," in *Proc. 9th Int. Conf. Rel., INFOCOM Technol. Optim.*, Sep. 2021, pp. 1–4.

[45] P. Thakur, K. Joshi, P. Thakral, and S. Jain, "Detection of email spam using machine learning algorithms: A comparative study," in *Proc. 8th Int. Conf. Signal Process. Commun. (ICSC)*, Dec. 2022, pp. 349–352.

[46] R. P. Cota and D. Zinca, "Comparative results of spam email detection using machine learning algorithms," in *Proc. 14th Int. Conf. Commun. (COMM)*, Jun. 2022, pp. 1–5.

[47] B. K. Dedeturk and B. Akay, "Spam filtering using a logistic regression model trained by an artificial bee colony algorithm," *Appl. Soft Comput.*, vol. 91, Jun. 2020, Art. no. 106229.

[48] Y. Kontsewaya, E. Antonov, and A. Artamonov, "Evaluating the effectiveness of machine learning methods for spam detection," *Proc. Comput. Sci.*, vol. 190, pp. 479–486, Jun. 2021.

[49] V. Sunjaya, S. Senjaya, J. Utama, H. Lucky, and D. Suhartono, "Content based spam classifying algorithms in email," *3rd Int. Conf. Artif. Intell. Data Sci.*, vol. 94, Jul. 2020, Art. no. 101716.

[50] T. Georgieva-Trifonova, "Research on filtering feature selection methods for e-mail spam detection by applying K-NN classifier," in *Proc. Int. Congr. Hum.-Comput. Interact., Optim. Robotic Appl. (HORA)*, Jun. 2022, pp. 1–4.

[51] L. N. Vejendla, B. Bysani, A. Mundru, M. Setty, and V. J. Kunta, "Score based support vector machine for spam mail detection," in *Proc. 7th Int. Conf. Trends Electron. Informat.*, 2023, pp. 915–920.

[52] H. Faris, F. A. Alqatawna, M. Al-Zoubi, and I. Aljarah. (2017). *Improving Email Spam Detection Using Content Based Feature Engineering Approach*. [Online]. Available: http://cran.r-project.org/web/packages/Boruta/index.html

[53] S. O. Olatunji, "Extreme learning machines and support vector machines models for email spam detection," in *Proc. IEEE 30th Can. Conf. Electr. Comput. Eng. (CCECE)*, Apr. 2017, pp. 1–6.

[54] X. Zheng, X. Zhang, Y. Yu, T. Kechadi, and C. Rong, "ELM-based spammer detection in social networks," *J. Supercomput.*, vol. 72, no. 8, pp. 2991–3005, Aug. 2016.

[55] S. Rezvani, X. Wang, and F. Pourpanah, "Intuitionistic fuzzy twin support vector machines," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 11, pp. 2140–2151, Nov. 2019.

[56] K. Juneja, "Two-phase fuzzy feature-filter based hybrid model for spam classification," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 10, pp. 10339–10355, Nov. 2022.

[57] I. Atacak, O. Çıtlak, and I. A. Dogru, "Application of interval type-2 fuzzy logic and type-1 fuzzy logic-based approaches to social networks for spam detection with combined feature capabilities," *PeerJ Comput. Sci.*, vol. 9, p. e1316, Apr. 2023.

[58] U. Srinivasarao and A. Sharaff, "SMS sentiment classification using an evolutionary optimization based fuzzy recurrent neural network," *Multimedia Tools Appl.*, vol. 82, no. 27, pp. 42207–42238, Nov. 2023.

[59] A. W. Wijayanto and Takdir, "Fighting cyber crime in email spamming: An evaluation of fuzzy clustering approach to classify spam messages," in *Proc. Int. Conf. Inf. Technol. Syst. Innov. (ICITSI)*, Nov. 2014, pp. 19–24.

[60] L. Bansal and N. Tiwari, "Feature selection based classification of spams using fuzzy support vector machine," in *Proc. Int. Conf. Smart Electron. Commun. (ICOSEC)*, Sep. 2020, pp. 258–263.

[61] S. Wang, X. Zhang, Y. Cheng, F. Jiang, W. Yu, and J. Peng, "A fast content-based spam filtering algorithm with fuzzy-SVM and k-means," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Jul. 2018, pp. 301–307.

[62] S. A. Khan, K. Iqbal, N. Mohammad, R. Akbar, S. S. A. Ali, and A. A. Siddiqui, "A novel fuzzy-logic-based multi-criteria metric for performance evaluation of spam email detection algorithms," *Appl. Sci.*, vol. 12, no. 14, p. 7043, Jul. 2022.

[63] X. Wang, Y. Zhao, and F. Pourpanah, "Recent advances in deep learning," *Int. J. Mach. Learn. Cybern.*, vol. 11, pp. 747–750, Jan. 2020.

[64] A. Kamilaris and F. X. Prenafeta-Boldu, "Deep learning in agriculture: A survey," *Comput. Electron. Agricult.*, vol. 147, pp. 70–90, Apr. 2018.

[65] L. Deng and D. Yu, "Deep learning: Methods and applications," *Found. Trends Signal Process.*, vol. 7, nos. 3–4, pp. 197–387, 2014.

[66] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, Apr. 2016.

[67] A. Baccouche, S. Ahmed, D. Sierra-Sosa, and A. Elmaghraby, "Malicious text identification: Deep learning from public comments and emails," *Information*, vol. 11, no. 6, p. 312, Jun. 2020.

[68] M. Alauthman, "Botnet spam e-Mail detection using deep recurrent neural network," *Int. J. Emerg. Trends Eng. Res.*, vol. 8, no. 5, pp. 1979–1986, May 2020.

[69] I. AbdulNabi and Q. Yaseen, "Spam email detection using deep learning techniques," *Proc. Comput. Sci.*, vol. 184, no. 2, pp. 853–858, 2021.

[70] A. A. Abdullahi and M. Kaya, "A deep learning based method to detect email and SMS spams," in *Proc. Int. Conf. Decis. Aid Sci. Appl. (DASA)*, Dec. 2021, pp. 430–435.

[71] K. F. Rafat, Q. Xin, A. R. Javed, Z. Jalil, and R. Z. Ahmad, "Evading obscure communication from spam emails," *Math. Biosciences Eng.*, vol. 19, no. 2, pp. 1926–1943, 2021.

[72] T. Wen, Y. Xiao, A. Wang, and H. Wang, "A novel hybrid feature fusion model for detecting phishing scam on Ethereum using deep neural network," *Expert Syst. Appl.*, vol. 211, Jan. 2023, Art. no. 118463.

[73] Z. Alom, B. Carminati, and E. Ferrari, "A deep learning model for Twitter spam detection," *Online Social Netw. Media*, vol. 18, Jul. 2020, Art. no. 100079.

[74] A. Makkar and N. Kumar, "An efficient deep learning-based scheme for Web spam detection in IoT environment," *Future Gener. Comput. Syst.*, vol. 108, pp. 467–487, Jul. 2020.

[75] S. Smadi, N. Aslam, and L. Zhang, "Detection of online phishing email using dynamic evolving neural network based on reinforcement learning," *Decis. Support Syst.*, vol. 107, pp. 88–102, Mar. 2018.

[76] S. Isik, Z. Kurt, Y. Anagun, and K. Ozkan, "Spam e-mail classification recurrent neural networks for spam e-mail classification on an agglutinative language," *Int. J. Intell. Syst. Appl. Eng.*, vol. 8, no. 4, pp. 221–227, Dec. 2020.

[77] H. Yang, Q. Liu, S. Zhou, and Y. Luo, "A spam filtering method based on multi-modal fusion," *Appl. Sci.*, vol. 9, no. 6, p. 1152, Mar. 2019.

[78] S. Gadde, A. Lakshmanarao, and S. Satyanarayana, "SMS spam detection using machine learning and deep learning techniques," in *Proc. 7th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, vol. 1, Mar. 2021, pp. 358–362.

[79] F. Wei and T. Nguyen, "A lightweight deep neural model for SMS spam detection," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, Oct. 2020, pp. 1–6.

[80] V. S. Vinitha, D. K. Renuka, and L. A. Kumar, "Long short-term memory networks for email spam classification," in *Proc. Int. Conf. Intell. Syst. Commun., IoT Secur. (ICISCoIS)*, Feb. 2023, pp. 176–180.

[81] S. Bagui, D. Nandi, S. Bagui, and R. J. White, "Machine learning and deep learning for phishing email classification using one-hot encoding," *J. Comput. Sci.*, vol. 17, no. 7, pp. 610–623, Jul. 2021.

[82] D. A. Otchere, T. O. A. Ganat, R. Gholami, and S. Ridha, "Application of supervised machine learning paradigms in the prediction of petroleum reservoir properties: Comparative analysis of ANN and SVM models," *J. Petroleum Sci. Eng.*, vol. 200, May 2021, Art. no. 108182.

[83] M. Mohammadi, T. A. Rashid, S. H. T. Karim, A. H. M. Aldalwie, Q. T. Tho, M. Bidaki, A. M. Rahmani, and M. Hosseinzadeh, "A comprehensive survey and taxonomy of the SVM-based intrusion detection systems," *J. Netw. Comput. Appl.*, vol. 178, Mar. 2021, Art. no. 102983.

[84] W. Wang, X. Du, D. Shan, and N. Wang, "A hybrid cloud intrusion detection method based on SDAE and SVM," in *Proc. 12th Int. Conf. Intell. Comput. Technol. Autom. (ICICTA)*, Oct. 2019, pp. 271–274.

[85] P. Navaney, G. Dubey, and A. Rana, "SMS spam filtering using supervised machine learning algorithms," in *Proc. 8th Int. Conf. Cloud Comput., Data Sci. Eng.*, Jan. 2018, pp. 43–48.

[86] D. Sculley and G. M. Wachman, "Relaxed online SVMs for spam filtering," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2007, pp. 415–422.

[87] P. Haider, U. Brefeld, and T. Scheffer, "Supervised clustering of streaming data for email batch detection," in *Proc. 24th Int. Conf. Mach. Learn.*, Jun. 2007, pp. 345–352.

[88] H. Zhou, J. Zhang, Y. Zhou, X. Guo, and Y. Ma, "A feature selection algorithm of decision tree based on feature weight," *Expert Syst. Appl.*, vol. 164, Feb. 2021, Art. no. 113842.

[89] M. M. Ghiasi, S. Zendehboudi, and A. A. Mohsenipour, "Decision tree-based diagnosis of coronary artery disease: CART model," *Comput. Methods Programs Biomed.*, vol. 192, Aug. 2020, Art. no. 105400.

[90] S. Rizvi, B. Rienties, and S. A. Khoja, "The role of demographics in online learning; a decision tree based approach," *Comput. Educ.*, vol. 137, pp. 32–47, Aug. 2019.

[91] A. Wijaya and A. Bisri, "Hybrid decision tree and logistic regression classifier for email spam detection," in *Proc. 8th Int. Conf. Inf. Technol. Electr. Eng. (ICITEE)*, Oct. 2016, pp. 1–4.

[92] A. F. Zulfikar, D. Supriyadi, Y. Heryadi, and Lukas, "Comparison performance of decision tree classification model for spam filtering with or without the recursive feature elimination (RFE) approach," in *Proc. 4th Int. Conf. Inf. Technol., Inf. Syst. Electr. Eng. (ICITISEE)*, Nov. 2019, pp. 311–316.

[93] Y. Zhang, S. Wang, P. Phillips, and G. Ji, "Binary PSO with mutation operator for feature selection using decision tree applied to spam detection," *Knowl.-Based Syst.*, vol. 64, pp. 22–31, Jul. 2014.

[94] H. Kaur and A. Sharma, "Improved email spam classification method using integrated particle swarm optimization and decision tree," in *Proc. 2nd Int. Conf. Next Gener. Comput. Technol. (NGCT)*, Oct. 2016, pp. 516–521.

[95] S. Uddin, I. Haque, H. Lu, M. A. Moni, and E. Gide, "Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction," *Sci. Rep.*, vol. 12, no. 1, p. 10358, Apr. 2022.

[96] H. Liu, J. An, W. Xu, X. Jia, L. Gan, and C. Yuen, "K-means based constellation optimization for index modulated reconfigurable intelligent surfaces," *IEEE Commun. Lett.*, vol. 27, no. 8, pp. 2152–2156, Jun. 2023.

[97] S. A. Orazbayev, R. E. Zhumadylov, A. T. Zhunisbekov, T. S. Ramazanov, and M. T. Gabdullin, "Obtaining of copper nanoparticles in combined RF+DC discharge plasma," *Mater. Today, Proc.*, vol. 20, pp. 329–334, Jun. 2020.

[98] D. Ö. Sahin and S. Demirci, "Spam filtering with KNN: Investigation of the effect of k value on classification performance," in *Proc. 28th Signal Process. Commun. Appl. Conf. (SIU)*, Oct. 2020, pp. 1–4.

[99] Y. K. Zamil, S. A. Ali, and M. A. Naser, "Spam image email filtering using K-NN and SVM," *Int. J. Electr. Comput. Eng. (IJECE)*, vol. 9, no. 1, p. 245, Feb. 2019.

[100] G. Hnini, J. Riffi, M. A. Mahraz, A. Yahyaouy, and H. Tairi, "Spam filtering system based on nearest neighbor algorithms," in *Proc. Int. Conf. Artif. Intell. Ind. Appl.*, 2021, pp. 36–46.

[101] A. S. Mashaleh, N. F. Binti Ibrahim, M. A. Al-Betar, H. M. J. Mustafa, and Q. M. Yaseen, "Detecting spam email with machine learning optimized with Harris hawks optimizer (HHO) algorithm," *Proc. Comput. Sci.*, vol. 201, pp. 659–664, Aug. 2022.

[102] M. Belgiu and L. Dragus, "Random forest in remote sensing: A review of applications and future directions," *ISPRS J. Photogramm. Remote Sens.*, vol. 114, pp. 24–31, Apr. 2016.

[103] J. L. Speiser, M. E. Miller, J. Tooze, and E. Ip, "A comparison of random forest variable selection methods for classification prediction modeling," *Expert Syst. Appl.*, vol. 134, pp. 93–101, Nov. 2019.

[104] N. R. Kothapally and V. Kakulapati, "Classification of spam messages using random forest algorithm," *J. Xidian University*, vol. 15, no. 8, pp. 495–505, 2021.

[105] A. Shrivastava and R. Dubey, "Classification of spam mail using different machine learning algorithms," in *Proc. Int. Conf. Adv. Comput. Telecommun. (ICACAT)*, Dec. 2018, pp. 1–10.

[106] K. L. Goh and A. K. Singh, "Comprehensive literature review on machine learning structures for web spam classification," *Proc. Comput. Sci.*, vol. 70, pp. 434–441, Jun. 2015.

[107] F. Ye, G. Chen, Q. Liu, L. Zhang, Q. Qi, B. Hu, and X. Fan, "A spam classification method based on naive Bayes," in *Proc. IEEE 6th Inf. Technol. Mechatronics Eng. Conf. (ITOEC)*, vol. 6, Mar. 2022, pp. 1856–1861.

[108] T. S. Guzella and W. M. Caminhas, "A review of machine learning approaches to spam filtering," *Expert Syst. Appl.*, vol. 36, no. 7, pp. 10206–10222, Sep. 2009.

[109] M. Sahami, S. Dumais, D. Heckerman, E. Horvitz, and G. Building, "A Bayesian approach to filtering junk e-mail," in *Proc. Learning Text Categorization, Workshop*, 1998, pp. 98–105.

[110] J. Kim, K. Chung, and K. Choi, "Spam filtering with dynamically updated URL statistics," *IEEE Secur. Privacy Mag.*, vol. 5, no. 4, pp. 33–39, Jul. 2007.

[111] X. Deng, Y. Li, J. Weng, and J. Zhang, "Feature selection for text classification: A review," *Multimedia Tools Appl.*, vol. 78, no. 3, pp. 3797–3816, Feb. 2019.

[112] A. Çıltık and T. Güngör, "Time-efficient spam e-mail filtering using n-gram models," *Pattern Recognit. Lett.*, vol. 29, no. 1, pp. 19–33, Jan. 2008.

[113] T. Toma, S. Hassan, and M. Arifuzzaman, "An analysis of supervised machine learning algorithms for spam email detection," in *Proc. Int. Conf. Autom., Control Mechatronics*, Jul. 2021, pp. 1–5.

[114] C. Li, G. Zhan, and Z. Li, "News text classification based on improved Bi-LSTM-CNN," in *Proc. 9th Int. Conf. Inf. Technol. Med. Educ. (ITME)*, Oct. 2018, pp. 890–893.

[115] H. Moayedi, M. Mehrabi, M. Mosallanezhad, A. S. A. Rashid, and B. Pradhan, "Modification of landslide susceptibility mapping using optimized PSO-ANN technique," *Eng. Comput.*, vol. 35, no. 3, pp. 967–984, Jul. 2019.

[116] A. Kurani, P. Doshi, A. Vakharia, and M. Shah, "A comprehensive comparative study of artificial neural network (ANN) and support vector machines (SVM) on stock forecasting," *Ann. Data Sci.*, vol. 10, no. 1, pp. 183–208, Feb. 2023.

[117] B. Ingre and A. Yadav, "Performance analysis of NSL-KDD dataset using ANN," in *Proc. Int. Conf. Signal Process. Commun. Eng. Syst.*, Jan. 2015, pp. 92–96.

[118] C. Zhan, F. Zhang, and M. Zheng, "Design and implementation of an optimization system of span filter rule based on neural network," in *Proc. Int. Conf. Commun., Circuits Syst.*, vol. 3, Jul. 2007, pp. 882–886.

[119] R. Talaei Pashiri, Y. Rostami, and M. Mahrami, "Spam detection through feature selection using artificial neural network and sine–cosine algorithm," *Math. Sci.*, vol. 14, no. 3, pp. 193–199, Sep. 2020.

[120] S. A. A. Ghaleb, M. Mohamad, E. F. H. S. Abdullah, and W. A. H. M. Ghanem, "Spam classification based on supervised learning using grasshopper optimization algorithm and artificial neural network," in *Proc. 2nd Int. Conf.*, 2021, pp. 420–434.

[121] A. Arram, H. Mousa, and A. Zainal, "Spam detection using hybrid artificial neural network and genetic algorithm," in *Proc. 13th Int. Conf. Intellient Syst. Design Appl.*, Dec. 2013, pp. 336–340.

[122] J. Gu, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, May 2018.

[123] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: Analysis, applications, and prospects," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022.

[124] V. Gupta, A. Mehta, A. Goel, U. Dixit, and A. C. Pandey, "Spam detection using ensemble learning," in *Harmony Search and Nature Inspired Optimization Algorithms: Theory and Applications*. Cham, Switzerland: Springer, 2019, pp. 661–668.
[125] M. Gupta, A. Bakliwal, S. Agarwal, and P. Mehndiratta, "A comparative study of spam SMS detection using machine learning classifiers," in *Proc. 11th Int. Conf. Contemp. Comput.*, Aug. 2018, pp. 1–7.
[126] M. Popovac, M. Karanovic, S. Sladojevic, M. Arsenovic, and A. Anderla, "Convolutional neural network based SMS spam detection," in *Proc. 26th Telecommun. Forum (TELFOR)*, Nov. 2018, pp. 1–4.
[127] T. Sharmin, F. Di Troia, K. Potika, and M. Stamp, "Convolutional neural networks for image spam detection," *Inf. Secur. J. A Global Perspective*, vol. 29, no. 3, pp. 103–117, May 2020.
[128] A. Farzad, H. Mashayekhi, and H. Hassanpour, "A comparative performance analysis of different activation functions in LSTM networks for classification," *Neural Comput. Appl.*, vol. 31, no. 7, pp. 2507–2521, Jul. 2019.
[129] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Phys. D, Nonlinear Phenomena*, vol. 404, Mar. 2020, Art. no. 132306.
[130] S. Muzaffar and A. Afshari, "Short-term load forecasts using LSTM networks," *Energy Proc.*, vol. 158, pp. 2922–2927, Feb. 2019.
[131] B. Lindemann, B. Maschler, N. Sahlab, and M. Weyrich, "A survey on anomaly detection for technical systems using LSTM networks," *Comput. Ind.*, vol. 131, Oct. 2021, Art. no. 103498.
[132] G. Jain, M. Sharma, and B. Agarwal, "Optimizing semantic lstm for spam detection," *Int. J. Inf. Technol.*, vol. 11, pp. 239–250, Jun. 2019.
[133] E. E. Eryilmaz, D. Ö. Sahin, and E. Kiliç, "Filtering Turkish spam using LSTM from deep learning techniques," in *Proc. 8th Int. Symp. Digit. Forensics Secur. (ISDFS)*, Jun. 2020, pp. 1–6.
[134] S. Thanarattananakin, S. Bulao, B. Visitsilp, and M. Maliyaem, "Spam detection using word embedding-based LSTM," in *Proc. Joint Int. Conf. Digit. Arts, Media Technol. ECTI Sect. Conf. Electr., Electron., Comput. Telecommun. Eng. (ECTI DAMT NCON)*, Jan. 2022, pp. 227–231.
[135] O. Yildirim, U. B. Baloglu, R.-S. Tan, E. J. Ciaccio, and U. R. Acharya, "A new approach for arrhythmia classification using deep coded features and LSTM networks," *Comput. Methods Programs Biomed.*, vol. 176, pp. 121–133, Jul. 2019.
[136] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," 2014, *arXiv:1409.1259*.
[137] P. T. Yamak, L. Yujian, and P. K. Gadosey, "A comparison between ARIMA, LSTM, and GRU for time series forecasting," in *Proc. 2nd Int. Conf. Algorithms, Comput. Artif. Intell.*, Dec. 2019, pp. 49–55.
[138] S. Gao, Y. Huang, S. Zhang, J. Han, G. Wang, M. Zhang, and Q. Lin, "Short-term runoff prediction with GRU and LSTM networks without requiring time step optimization during sample generation," *J. Hydrol.*, vol. 589, Oct. 2020, Art. no. 125188.
[139] K. A. Al-Thelaya, T. S. Al-Nethary, and E. Y. Ramadan, "Social networks spam detection using graph-based features analysis and sequence of interactions between users," in *Proc. IEEE Int. Conf. Informat., IoT, Enabling Technol. (ICIoT)*, Feb. 2020, pp. 206–211.
[140] T. Repke and R. Krestel. (2018). *Bringing Back Structure To Free Text Email Conversations With Recurrent Neural Networks*. [Online]. Available: http://isc.enron.com/
[141] T. Le, M. Vo, B. Vo, E. Hwang, S. Rho, and S. Baik, "Improving electric energy consumption prediction using CNN and bi-LSTM," *Appl. Sci.*, vol. 9, no. 20, p. 4237, Oct. 2019.
[142] F. Shahid, A. Zameer, and M. Muneeb, "Predictions for COVID-19 with deep learning models of LSTM, GRU and bi-LSTM," *Chaos, Solitons Fractals*, vol. 140, Nov. 2020, Art. no. 110212.
[143] S. M. Zaman, M. M. Hasan, R. I. Sakline, D. Das, and M. A. Alam, "A comparative analysis of optimizers in recurrent neural networks for text classification," in *Proc. IEEE Asia–Pacific Conf. Comput. Sci. Data Eng. (CSDE)*, vol. 3, Dec. 2021, pp. 1–6.
[144] S. E. Rahman and S. Ullah, "Email spam detection using bidirectional long short term memory with convolutional neural network," in *Proc. IEEE Region 10 Symp. (TENSYMP)*, Jun. 2020, pp. 1307–1311.
[145] C. M. Shaik, N. M. Penumaka, S. K. Abbireddy, V. Kumar, and S. S. Aravinth, "Bi-LSTM and conventional classifiers for email spam filtering," in *Proc. 3rd Int. Conf. Artif. Intell. Smart Energy (ICAIS)*, Feb. 2023, pp. 1350–1355.
[146] A. L. Rosewelt, N. D. Raju, and S. Ganapathy, "An effective spam message detection model using feature engineering and bi-LSTM," in *Proc. Int. Conf. Adv. Comput., Commun. Appl. Informat. (ACCAI)*, Jan. 2022, pp. 1–6.
[147] Y. Gao, M. Yang, X. Zhao, B. Pardo, Y. Wu, T. N. Pappas, and A. Choudhary, "Image spam hunter," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, vol. 2, Mar. 2008, pp. 1765–1768.
[148] M. Dredze, R. Gevaryahu, and A. Elias-Bachrach. (2007). *Learning Fast Classifiers for Image Spam*. [Online]. Available: http://fuzzyocr.own-hero.net/
[149] Z. Wang, W. Josephson, Q. Lv, M. Charikar, and K. Li, "Filtering image spam with near-duplicate detection," in *Proc. CEAS*, 2007, pp. 1–10.
[150] D. Debarr and H. Wechsler. (2007). *Spam Detection Using Clustering, Random Forests, and Active Learning*. [Online]. Available: http://trec.nist.gov/pubs/trec16/papers/SPAM.OVERVIEW1
[151] I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C. D. Spyropoulos, and P. Stamatopoulos. (2006). *Learning to Filter Spam e-mail: A Comparison of a Naive Bayesian and a Memory-based Approach 1*. [Online]. Available: http://www.cauce.org
[152] I. Koprinska, J. Poon, J. Clark, and J. Chan, "Learning to classify e-mail," *Inf. Sci.*, vol. 177, no. 10, pp. 2167–2187, May 2007.
[153] B. Biggio, I. Corona, G. Fumera, G. Giacinto, and F. Roli, "Bagging classifiers for fighting poisoning attacks in adversarial classification tasks," in *Proc. 10th Int. Workshop*, 2011, pp. 350–359.
[154] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, G. Paliouras, and C. D. Spyropoulos. (2000). *An Evaluation of Naive Bayesian Anti-Spam Filtering*. [Online]. Available: http://www.caucee.org
[155] G. V. Cormack and T. R. Lynam, "Online supervised spam filter evaluation," *ACM Trans. Inf. Syst.*, vol. 25, no. 3, p. 11, Jul. 2007.
[156] L. Zhang, J. Zhu, and T. Yao, "An evaluation of statistical spam filtering techniques," *ACM Trans. Asian Lang. Inf. Process.*, vol. 3, no. 4, pp. 243–269, Dec. 2004.
[157] J. R. Mendez, F. Fdez-Riverola, F. Dāsaz, E. L. Iglesias, and J. M. Corchado, "A comparative performance study of feature selection methods for the anti-spam filtering domain," in *Proc. 6th Ind. Conf. Data Mining*, 2006, pp. 106–120.
[158] A. Attar, R. M. Rad, and R. E. Atani, "A survey of image spamming and filtering techniques," *Artif. Intell. Rev.*, vol. 40, no. 1, pp. 71–105, Jun. 2013.
[159] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. D. Spyropoulos, and P. Stamatopoulos. (2001). *Stacking Classifiers for Anti-spam Filtering of e-mail*. [Online]. Available: www.junkemail.org
[160] T. A. Almeida and A. Yamakami, "Content-based spam filtering," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2010, pp. 1–7.
[161] D. Mallampati and N. P. Hegde, "Feature extraction and classification of email spam detection using IMTF-IDF+skip-thought vectors," *Ingenierie Des. Syst. Inf.*, vol. 27, no. 6, pp. 941–948, Dec. 2022.
[162] H. Saini and K. S. Saini, "Hybrid model for email spam prediction using random forest for feature extraction," in *Proc. Int. Conf. Artif. Intell. Appl. (ICAIA) Alliance Technol. Conf. (ATCON-1)*, Apr. 2023, pp. 1–4.
[163] Q. Cheng, A. Xu, X. Li, and L. Ding, "Adversarial email generation against spam detection models through feature perturbation," in *Proc. IEEE Int. Conf. Assured Autonomy (ICAA)*, Mar. 2022, pp. 83–92.
[164] M. A. Hassan and N. Mtetwa, "Feature extraction and classification of spam emails," in *Proc. 5th Int. Conf. Soft Comput. Mach. Intell. (ISCMI)*, Nov. 2018, pp. 93–98.
[165] I. Inuwa-Dutse, M. Liptrott, and I. Korkontzelos, "Detection of spam-posting accounts on Twitter," *Neurocomputing*, vol. 315, pp. 496–511, Nov. 2018.
[166] S. Aiyar and N. P. Shetty, "N-gram assisted YouTube spam comment detection," *Proc. Comput. Sci.*, vol. 132, pp. 174–182, Jul. 2018.
[167] R. Alharthi, A. Alhothali, and K. Moria, "A real-time deep-learning approach for filtering Arabic low-quality content and accounts on Twitter," *Inf. Syst.*, vol. 99, Jul. 2021, Art. no. 101740.
[168] Y. Liu, B. Pang, and X. Wang, "Opinion spam detection by incorporating multimodal embedded representation into a probabilistic review graph," *Neurocomputing*, vol. 366, pp. 276–283, Nov. 2019.
[169] T. Wu, S. Liu, J. Zhang, and Y. Xiang, "Twitter spam detection based on deep learning," in *ACM Int. Conf. Proc. Ser.*, Jan. 2017, pp. 1–26.
[170] A. Barushka and P. Hajek, "Review spam detection using word embeddings and deep neural networks," in *Proc. 15th IFIP WG*, 2019, pp. 340–350.
[171] I. Kanaris, K. Kanaris, and E. Stamatatos, "Spam detection using character n-grams," in *Proc. 4th Helenic Conf.*, 2006, pp. 95–104.

EKRAMUL HAQUE TUSHER received the B.Sc. degree in computer science from International Islamic University Chittagong (IIUC). He is currently pursuing the master's degree in soft computing and intelligent systems with Universiti Malaysia Pahang Al-Sultan Abdullah (UMPSA), Pekan, Pahang, Malaysia. He has been a Research Assistant with the Machine Intelligence Research Group (MIRG), UMPSA, since 2023. His current research interests include machine learning methods, deep learning, fuzzy systems, and explainable AI.

MOHD ARFIAN ISMAIL received the B.Sc., M.Sc., and Ph.D. degrees in computer science from Universiti Teknologi Malaysia (UTM), in 2008, 2011, and 2016, respectively. He is currently an Associate Professor with the Faculty of Computing, University Malaysia Pahang Al-Sultan Abdullah, Malaysia. His current research interests include machine learning methods and fuzzy systems.

MD ARAFATUR RAHMAN (Senior Member, IEEE) received the Ph.D. degree in electronic and telecommunications engineering from the University of Naples Federico II, Naples, Italy, in 2013. He has around 15 years of research and teaching experience in the domain of computer science and communications engineering. He was an Associate Professor with the Faculty of Computing, Universiti Malaysia Pahang. He was a Postdoctoral Research Fellow with the University of Naples Federico II, in 2014, and a Visiting Researcher with the Sapienza University of Rome, in 2016. Currently, he is a Reader of cyber security with the School of Engineering, Computing and Mathematical Sciences, University of Wolverhampton, U.K. He has developed an excellent track record of academic leadership and management and execution of international ICT projects that are supported by agencies in the U.K., Italy, EU, and Malaysia. He has co-authored around 150 prestigious IEEE and Elsevier journals, such as IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, IEEE TRANSACTIONS ON SERVICES COMPUTING, *IEEE Communications Magazine*, JNCA (Elsevier), and FGCS (Elsevier); and conference publications, such as IEEE Globecom and IEEE DASC. His research interests include cyber security, in particular on the Internet of Things (IoT), wireless communication networks, cognitive radio networks, 5G, vehicular communication, cyber-physical systems, big data, cloud-fog-edge computing, and machine learning-dependent applications. He was a fellow of the IBM Center of Excellence and the Earth Resources and Sustainability Center, Malaysia. He was endorsed by the Royal Academy of Engineering, U.K., as a Global Talent under the category of exceptional talent, in 2022. He was awarded the Higher Education Academy (HEA) Fellowship from the U.K. He has received several prestigious international research awards, notably the Best Paper Award at ICNS'15 (Italy); IC0902 Grant (France); Italian Government Ph.D. Research Scholarship; the IIUM Best Masters Student Award; the Best Supervisor Award at UMP; and the Awards in International Exhibitions, including the Euro Business-HALLER Poland Special Award at MTE 2022; the Best Innovation Award at MTE 2020, Malaysia; the Diamond and Gold in BiS'17 U.K.; the Best of the Best Innovation Award and Most Commercial IT Innovation Award, Malaysia; and the Gold and Silver Medals in iENA'17 Germany. He served as the Specialty Chief Editor for IoT Theory and Fundamental Research (specialty section of *Frontiers in the Internet of Things*); an Advisory Board Member and an Editorial Board Member for *Computer Systems Science and Engineering* (Tech Science Press) and *Computers* (MDPI); a Lead Guest Editor for IEEE ACCESS and *Computers*; an Associate Editor for IEEE ACCESS and *Patron*; the General Chair; the Organizing Committee Member; the Publicity Chair; the Session Chair; the Programme Committee Member; and a member of the Technical Programme Committee (TPC) in numerous leading conferences worldwide, such as IEEE Globecom, IEEE DASC, IEEE iSCI, and IEEE ETCCE, and journals. His name was enlisted inside the World Top 2% Scientists list released by Stanford University under the category of Citation Impact in Single Calendar Year in 2019, 2020, and 2021.

ALI H. ALENEZI received the B.S. degree in electrical engineering from King Saud University, Saudi Arabia, the M.S. degree in electrical engineering from the KTH Royal Institute of Technology, Sweden, and the Ph.D. degree in electrical engineering from New Jersey Institute of Technology, USA, in 2018. He is currently an Associate Professor with the Electrical Engineering Department, Northern Border University, Saudi Arabia. His research interests include acoustic communication, wireless communications, and 4G and 5G networks using UAVs.

MUEEN UDDIN (Senior Member, IEEE) received the Ph.D. degree from Universiti Teknologi Malaysia (UTM), in 2013. He is currently an Associate Professor of data and cybersecurity with the University of Doha for Science and Technology, Qatar. He has published more than 130 international journals and conference papers in highly reputed journals with a cumulative impact factor of over 300. His research interests include blockchain, cybersecurity, the IoT security, and network and cloud security.